ORIGINAL

KARL J. KRAMER (No. 136433)
JANA G. GOLD (No. 154246)
MORRISON & FOERSTER LLP
755 Page Mill Road
Palo Alto, California 94304-1018
Telephone: (415)813-5600

RAOUL D. KENNEDY (No. 40892)
MORRISON & FOERSTER LLP
345 California Street
San Francisco, California 94104-2675
Telephone: (415) 677-7000

PATRICK J. FLINN (No. 104423)
ALSTON & BIRD
One Atlantic Center
1201 W. Peachtree Street
Atlanta, Georgia 30306
Telephone: (404)881-7000

Attorneys for Defendants/Counter-Claimants CYLINK CORPORATION, CARO-KANN CORPORATION and STANFORD UNIVERSITY

161

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

ROGER SCHLAFLY,

Plaintiff,

v.

PUBLIC KEY PARTNERS AND RSA DATA SECURITY, INC.,

Defendants,

No. C-94-20512 SW

RSA DATA SECURITY, INC.,

Plaintiff,

v.

CYLINK CORPORATION and CARO-KANN CORPORATION, et al.

Defendants.

No. C-96-20094 SW

DEFENDANTS' MOTION FOR SUMMARY JUDGMENT ON THE VALIDITY OF THE HELLMAN-MERKLE PATENT

Date: September 4, 1996
Time: 10:00 a.m.
Courtroom: 4

Hon. Spencer Williams

NOTICE OF MOTION AND MOTION

Please take notice that on September 4, 1996, at 10:00 a.m., or as soon thereafter as the matter may be heard, defendants and counter-claimants Cylink Corporation ("Cylink"), Caro-Kann Corporation ("Caro-Kann") and The Board of Trustees of the Leland Stanford Junior University ("Stanford") (collectively "defendants") will move, and hereby do move, for summary judgment that RSA Data Security, Inc. ("RSADSI") cannot prove by clear and convincing evidence that the claims of U.S. Patent No. 4,218,582 (the "Hellman-Merkle Patent") are invalid.

MEMORANDUM OF POINTS AND AUTHORITIES

Introduction

RSADSI cannot meet its burden of proving invalidity by clear and convincing evidence. RSADSI cannot provide a single piece of prior art purported to invalidate the Hellman-Merkle Patent. Indeed, RSADSI apparently raises just one invalidity argument: that the claimed inventions were not enabled. RSADSI's non-enablement argument, however, contradicts well-established and controlling precedent. Because enablement is a question of law for the Court to determine, RSADSI's argument can and should be disposed of without further delay.[1]

The claims of the Hellman-Merkle patent require mathematical operations that are "computationally infeasible" to invert. RSADSI argues that the claims were not enabled because, after the patent issued, mathematicians devised "feasible" methods to invert certain

---

[1] This defense was raised and briefed extensively during the Motion for Preliminary Injunction. The Court's Order on the Preliminary Injunction Motion did not address it.

mathematical operations disclosed in the patent specification. Although RSADSI's account of the facts is inaccurate, it is not necessary to debate those facts for the purposes of this motion. Even if the facts were exactly as RSADSI claims them to be, RSADSI's argument would fail as a matter of law. Enablement is measured at the time the patent application was filed. RSADSI presents no evidence that the claims were not enabled at the time the patent application was filed. Indeed, RSADSI's founder, director and leading scientist, Professor Rivest, admitted that the inventions were enabled at the time of filing:

> At the time of the filing, all three systems [disclosed in the Hellman-Merkle Patent] were, as far as I know, "computationally infeasible" according to their definition.

[Exh. A at 6; see also Exh. B]

RSADSI cannot overcome the presumed validity of the Hellman-Merkle Patent claims. Summary judgment should be entered against all of RSADSI's allegations that the Hellman-Merkle Patent claims are invalid.

Background Facts

The Hellman-Merkle patent application was filed on October 6, 1977 and granted by the United States Patent Office on August 19, 1980. [Exh. C hereto] The Hellman-Merkle Patent claims a new cryptography system, called public-key cryptography, which was invented at Stanford University. David Kahn, the preeminent historian of cryptography, describes the Stanford invention as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance." [Exh. D]

/ / /

Since the creation of the first secret codes thousands of years ago, a single "key" has been used both to encode and decode a message. See D. Kahn, The Codebreakers (1967). The key had to be exchanged secretly, perhaps in person or over insecure channels, before coded communications could begin. [Id.] Such systems are inherently vulnerable because of the risk that the key will be intercepted by someone who can eavesdrop on the subsequent communications. Thus, as in innumerable fictional and actual military struggles, life and death often depended upon espionage operations designed to intercept or crack the shared secret key. [Id.] The invention of public key cryptography at Stanford makes all of those efforts irrelevant.

In public-key cryptography, a person wishing to receive coded messages generates two related numbers: a secret decoding key, which is shared with no one, and a public encoding key, which may be shared with anyone or everyone in the world. [See, e.g., Exh. C, 2:38-59] The keys are mathematically related such that the secret key is "computationally infeasible" to derive solely from the public key.[2] [Id.] Anyone wishing to send a coded message uses the person's public key to encode the message. The recipient decodes the message using the corresponding secret key that was never shared with anyone. Because the mathematical operation that was used to encode the message is "computationally infeasible" to invert using just the public key, only the secret private key can decode the

---

[2] The patent states that "[a] task is considered computationally infeasible if its cost as measured by either the amount of memory used or the computing time is finite but impossibly large, for example, on the order of approximately 10^{30} operations with existing computational methods and equipment." [Exh. C, 5:10-14]

message. [Id.] Thus, at the heart of public-key cryptography is an elegant yet completely counter-intuitive approach: the encoding key can be sent directly to the enemy without compromising the security of the messages encoded with the key.

In RSADSI's own words, the Stanford inventors "invented public-key cryptography" (Exh. E (emphasis supplied)), and the Stanford Patents, including the Hellman-Merkle Patent contain "[t]he basic ideas of public-key cryptography. . . " (Exh. F). RSADSI's founder, Professor Rivest, has admitted that the Hellman-Merkle Patent was enabling. [Exh. A at 6]

ARGUMENT

I. SUMMARY JUDGMENT SHOULD BE GRANTED WHEN, AS A MATTER OF LAW, THE CHALLENGER CANNOT PROVE BY CLEAR AND CONVINCING EVIDENCE THAT THE PATENT IS INVALID.

The Hellman-Merkle Patent claims are presumed to be valid as a matter of law. 35 U.S.C. § 282. To overcome this presumption, RSADSI carries the burden of proving invalidity by clear and convincing evidence. See, e.g., American Hoist & Derrick Co. v. Sowa & Sons, Inc., 725 F.2d 1350, 1360 (Fed. Cir.), cert. denied, 469 U.S. 821, 105 S.Ct. 95, 83 L. Ed. 2d 41 (1984). The presumption of validity shifts "the burden of going forward as well as the burden of proof of facts to the challenger." Avia Group Int'l, Inc. v. L.A. Gear California, 853 F.2d 1557, 1562 (Fed. Cir. 1988) (affirming summary judgment that defendant failed to prove patent invalidity).

Summary judgment should be granted if there are no genuinely disputed issues of material fact. Fed. R. Civ. P. 56. Because RSADSI carries the burden of proving the Hellman-Merkle Patent claims invalid, RSADSI's failure to raise a genuine issue of

material fact requires entry of summary judgment in favor of the defendants. Celotex Corp. v. Catrett, 477 U.S. 317, 322-23, 106 S.Ct. 2548, 2552-53, 91 L.Ed.2d 265 (1986). "[A] nonmovant must do more than merely raise some doubt as to the existence of a fact; evidence must be forthcoming from the nonmovant which would be sufficient to require submission to the jury of the dispute over the fact." Avia Group, 853 F.2d at 1560. "If the evidence [of the nonmovant] is merely colorable, or is not significantly probative, summary judgment may be granted." Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 249-50, 106 S.Ct. 2505, 2511, 91 L.Ed. 2d 202 (1986). RSADSI's evidence of invalidity is neither colorable nor probative.[3]

II. AS A MATTER OF LAW, RSADSI CANNOT CARRY ITS BURDEN OF PROVING BY CLEAR AND CONVINCING EVIDENCE THAT THE HELLMAN-MERKLE PATENT CLAIMS ARE INVALID.

RSADSI's validity challenge to the Hellman-Merkle Patent relies upon the enablement requirement of 35 U.S.C. § 112. RSADSI asserts that the Hellman-Merkle invention was not enabled because "trapdoor knapsack" algorithms disclosed in the patent were eventually solved, or "broken," by mathematicians and thus were not actually "computationally infeasible" as required by the claims. Enablement is an issue of law for the Court to decide. Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 941 (Fed. Cir.), cert. denied, 498 U.S. 920 (1990).

/ / /

/ / /

---

[3] Because a patent is presumed valid, the patentee has "no obligation to introduce any evidence initially on validity." Avia Group, 853 F.2d at 1562.

A. Enablement Is Tested As Of The Patent Filing Date.

RSADSI's assertion that the algorithms were broken well after the patent application was filed is irrelevant as a matter of law. It has long been recognized that enablement "is determined as of the filing date of the patent application." Hybritech, Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367, 1384 (Fed. Cir. 1986), cert. denied, 480 U.S. 947 (1987). If a patent disclosure is enabling as of the filing date, subsequent developments in the art cannot render it non-enabling. In re Hogan, 559 F.2d 595, 605 (C.C.P.A. 1977).[4]

In Hogan, the applicant disclosed a crystalline form of a new form of polymer, but claimed broadly all forms of the new polymer. Id. at 598-600. The Patent Office rejected the broad claims, citing later developed amorphous forms of the polymer that clearly were not enabled by the patent disclosure. The Court held that if the claims were enabled when the application was filed, then "enablement was established for all time and a later change in the state of the art cannot change it." Id. at 605. The Court concluded:

> Consideration of a later existing state of the art in testing for compliance with § 112, first paragraph, would not only preclude the grant of broad claims, but would wreak havoc in other ways as well. The use of a subsequently-existing improvement to show lack of enablement in an earlier-filed application on the basic invention would preclude issuance of a patent to the inventor of the thing improved, and in the case of issued patents, would invalidate all claims. . . .

---

[4] See also United States Steel Corp. v. Phillips Petroleum Co., 865 F.2d 1247, 1251 (Fed. Cir. 1989) (sufficiency of patent disclosure "must be judged as of the filing date"); Ares-Serono, Inc. v. Organon Int'l B.V., 862 F. Supp. 603, 606-7 (D. Mass. 1994) (tests after the patent filing are irrelevant to enablement).

Id. at 606. Thus, as a matter of law, a patent is enabled if a person skilled in the art could, as here, practice the invention as of the date the patent application was filed.

B. RSADSI Presents No Material Evidence On Enablement.

RSADSI's evidence is legally insufficient to raise a genuine issue of material fact. RSADSI's expert, Dr. Konheim, merely asserts, in the present tense, that the claims are not now enabled: "the '582 fails to disclose or teach one how to make a public key system which meets the requirements of claims 1 and 6. [Exh. G (Konheim Decl. ¶ 6)][5] For his conclusion, Dr. Konheim relies upon later developed methods of factoring, or solving, the knapsack algorithms disclosed in the patent. [Id. ¶¶ 6, 24 and Exhibits 1-4 thereto]

RSADSI presents no evidence that the disclosed operations were computationally "feasible" when the Hellman-Merkle Patent application was filed. Indeed, even to make that assertion, RSADSI would have to contradict its founder and director, Professor Rivest, who concluded in 1983 when reviewing the validity of the Hellman-Merkle patent claims: "[a]t the time of the filing, all three systems [disclosed in the Hellman-Merkle Patent] were, as far as I know, 'computationally infeasible' according to their definition." [Exh. A at 6] That fact is conclusive, for, as a matter of law, enablement is measured at the time the patent application was filed. Phillips Petroleum,

---

[5] This declaration, filed in response to the Preliminary Injunction Motion, is submitted again for the Court's convenience.

865 F.2d at 1251-2; Hybritech, 802 F.2d at 1384; Hogan, 559 F.2d at 605-06; Ares-Serono, 862 F. Supp. at 606-7.

CONCLUSION

RSADSI cannot escape the undisputed facts that compel summary judgment on its invalidity allegations. The only defense RSADSI has articulated with respect to the Hellman-Merkle Patent, enablement, is deficient as a matter of law. Summary judgment should be entered against RSADSI and in favor of the defendants as to those claims and defenses.

Dated: July 31, 1996

MORRISON & FOERSTER LLP
ALSTON & BIRD

By: _____
Karl J. Kramer

Attorneys for Defendants/ Counter-Claimants CYLINK CORPORATION, CARO-KANN CORPORATION and STANFORD UNIVERSITY

4

Notes on the Validity of Stanford Patent 4,218,582

Ronald L. Rivest

August 10, 1983

The purpose of this note is to review (from my layman's viewpoint) the validity of Stanford's Patent 4,218,582, with a view towards preparing a case that a license from this patent is not needed in order to practice the RSA cryptosystem. Of course, this discussion may cover points that are legally irrelevant, or present viewpoints that represent a misinterpretation of the patent law, since I am not a lawyer. Nonetheless, it is hoped that this note will be of value should it be necessary to prepare at some point a proper legal case.

Although the patent has 17 claims, only the first six claims are conceivably relevant. Claims 7-17 relate specifically to "knapsack"-based public-key cryptosystems, and so are irrelevant to the potential user of the RSA cryptosystem. However, claims 1 to 6 would seem to cover the broad aspects of "public-key cryptography" as it has become known in the literature:

claim 1: generating matched public/secret keys at the receiver, sending the public key to the transmitter, who encrypts the message with the public key and sends it back to the receiver, who deciphers it.

claim 2: (Dependent on claim 1) Authenticating identity of receiver by his ability to decipher message.

claim 3: (Dependent on claim 2) Having receiver prove his identity to transmitter by sending back the correctly deciphered message.

claim 4: generating digital signatures by generating secret/public keys at the transmitter, sending public key to receiver, encrypting message with secret key to get signature, sending both message and signature to receiver, who validates signature by checking that public key applied to signature yields message.

claim 5: Similar to 4, except only signature (not message) is sent. Message must have redundancy.

Note on the Validity of Stanford Patent 4,218,582

Ronald L. Rivest August 10, 1983

claim 6: An apparatus claim otherwise identical to claim 1. (Note that claims 1-5 are method claims.)

My comments here are based on a review of the U.S. Patent Office file on this patent, obtained for me by the MIT Patent Office. The major events in the file are:

6 Oct 77: Initial filing date (with 19 claims).

14 Nov 78: First Patent Office Action. All claims rejected or objected to, on grounds that

(1) The Diffie-Hellman paper "Multiuser Cryptographic techniques" (published June 1976), "teaches public key encryption technique, with an illustrative workable algorithm example, which meets the claim limitations".

(2) Claims are obvious over "Multiuser..." paper.

(3) Object to phrases "easy to effect but difficult to invert" in claims as only relative and failing to "point out and distinctly claim the invention."

(4) Claims do not appear distinct from earlier patent application by Merkle and Hellman (public-key distribution technique).

Some of the knapsack claims were deemed allowable, if they were rewritten so as not to be dependent on rejected claims.

2 Feb 79: Stanford's response. Rewrote knapsack claims to be independent of general public-key claims. Rewrote general public-key claims to include claim that secret deciphering key is computationally infeasible to generate from public enciphering key. Argued that "Multiuser..." paper did not teach art as now claimed, since the examples given there were not regarded as being difficult to break. Argued that earlier patent application (public-key distribution scheme) was very different in content.

Notes on the Validity of Stanford Patent 4,218,582

Ronald L. Rivest August 10, 1983

7 May 79: Second Patent Office Action. The general public key claims were rejected as being obvious over the "Multiuser..." paper, in view of the material presented in the Evans/Kantrowitz/Weiss paper, "A User Authentication Scheme Not Requiring Secrecy in the Computer" (CACM August 1974), a new reference included in the Stanford response of 2 Feb. "While the applicants' arguments point to the failure of the Diffie et al article originally cited to sufficiently disclose a workable function within the framework of appliants' definition of "computationally infeasible to invert", it is the position of the examiner that the Evans, Jr. et al article, cited by the applicants, does provide sufficient teachings of such functions, which would be obvious to implement in a public key system." Allowed some more knapsack claims and said other knapsack claims would be allowed if rewritten so as to be independent of rejected general public-key claims.

4 Oct 79: Response by Stanford to Second Patent Office Action. Rewrite general public-key claims to base generation of secret and public keys on random number generated as needed. Rewrite knapsack claims as independent claims so they would be allowed as previously stated by examiner. Argument that with keys generated by random numbers, the claims do not follow by any combined reading of "Multiuser..." plus "A User...". Argue, thus, that the combination of these prior articles "fails to teach under 35 U.S.C. 102 or 103."

13 Nov 79: Third Patent Office Action. All pending claims allowed, with statement that reason for allowance is that generating public key from random numbers distinguishes claims over prior art of Evans et. al.

(End of file)

Notes on the Validity of Stanford Patent 4,218,582

Ronald L. Rivest August 10, 1983

Discussion of possible weaknesses:

(1) Lack of inventiveness over prior art.

Claims 1-6 of the patent do not, in my opinion, say anything new over what is contained in the Diffie-Hellman paper, "Multiuser cryptographic techniques". Since this paper was published more than one year prior to the filing date of the patent, one might be able to argue that claims 1-6 are invalid. This argument may not be simple, however, since the file does contain a rather extensive discussion of this paper and its relation to claims 1-6. Stanford's primary point is that their claims (as amended) claim only methods in which the secret key is computationally infeasible to generate from the public key. While this point is stated in the "Multiuser..." paper (page 110, column 2, line 36), none of the proposed implementations of the public-key concept meet this criterion, according to Stanford. Further, the paper states that the public key concept "requires further work before it becomes implementable" (page 109, column 2, next to last line), and that "At present, we have neither a proof that public-key systems exist, nor a demonstration system." (page 111, column 1, line 1). The paper also states, "While the above arguments provide plausibility as opposed to proof, we hope they will stimulate additional work on this promising area of research." (page 111, column 2, line 36.) Stanford thus would be prepared to argue that the "Multiuser..." paper does not represent prior invention because it "fails to teach" a prospective user how to implement the public-key concept.

If the crux of the matter is the notion of "computational infeasibility", then a brief digression on this point is warranted. According to the specifications of the patent:

"A task is considered computational infeasible if its cost as measured by either the amount of memory used or the computing time is finite but impossibly large, for example.

on the order of approximately 10^{30} operations with existing computational methods and equipment." (column 5, line 10). I find this definition to be bizarre, for the reason that it depends on the current state of computer science. There is certainly some "truth" about the computational difficulty of some problem: it is either "easy" or it is not. If it is easy, then there exists some computer algorithm for solving the problem quickly. If it is not easy, then there is no such algorithm. Independent of what humans may know about these matters, each problem can be characterized as being either "easy" or "computationally infeasible". As we discover new algorithms, problems which were previously thought to be computationally infeasibly are discovered to be easy. As human knowledge progresses, more and more problems are discovered to be easy, and fewer and fewer remain as those believed to be "computationally infeasible".

Apparently the patent office may have granted claims 1-6 on the grounds that claims 7-17 (the knapsack system) provided a demonstration public-key system that was "computationally infeasible" as defined in the specifications. This example system, together with the restriction in the claims to systems that were "computationally infeasible" to break, apparently satisfied the examiner that something new was being claimed, that had not been taught in the prior art.

Curiously enough, the bizarre nature of the definition of "computationally infeasible" has turned out to be quite relevant. The knapsacks claims cover three related, but technically distinguishable, proposals. Let's denote them by B, B', and L. Here B is the "basic" knapsack system, exemplified by claim 7, B' is a simple variation on it, exemplified by claim 8, and L is a knapsack system based on discrete logarithms, exemplified by claim 9. (Claims 10-17 are either rather general claims covering all three

knapsack schemes (e.g. claim 13), or are "apparatus" claims corresponding to the "method" claims previously noted.)

At the time of filing, all three systems were, as far as I know, "computationally infeasible" according to their definition. Now, however, system B has been "broken" -- it is no longer computationally infeasible. A new algorithm has been discovered (by Adi Shamir) that shows how to compute the secret key for this system from the public key. A major portion of Stanford's justification for the inventiveness of claims 1-6 over the prior art has vanished into thin air! While the exact status of systems B' and L is (as far as I know) still open, there has been such a rush of progress on breaking knapsack systems in the past few years that I doubt that any researcher in the area (except, perhaps, Hellman or Merkle) would bet more than $10 even odds that systems B' or L will last out the next five years.

We see, thus, the nonsensical nature of the proposed definition of "computational infeasibility". Suppose, for example, that Shamir's paper had broken all three versions of the knapsack system as claimed. Would claims 1-6 still be accepted as valid? I would hope not. Essentially, they would be able to claim that a computational problem is hard because no one had thought about it (hard enough) before. It is not "infeasible" to build a swimming pool on the moon, just because it has not been done yet.

In a similar vein, one could argue that one of the proposed systems defined in "Multiuser..." was in fact "computationally infeasible" as defined in the specifications of the patent. I am referring to the system based on random transformations of circuit schematics, given on page 111 (paragraph beginning at the bottom of column 1). It is perhaps curious that the arguments made by Stanford in their remarks of 2 Feb 79 and 4 Oct 79 do not refer to this "circuit

schematic" proposal, but only to the "matrix inversion" proposal. Yet, the "Multiuser..." paper notes that the matrix inversion systems lacks "practical utility" (page 111, column 1, line 37), and then proceeds in a "more promising direction" (page 111, column 1, line 40) to the circuit schematic idea, which "appears more promising" (page 111, column 1, line 54). One can easily argue that Stanford has failed to supply sufficient evidence why the "Multiuser..." paper does not represent prior art, even with the claims amended to include reference to "computational infeasibility". Their arguments only refer to the matrix inversion example, which is admittedly weak. In order to complete their argument, however, it would seem necessary to argue that all the systems proposed in the "Multiuser..." paper fail to meet the "computational infeasibility" requirement as claimed, and not just the matrix inversion example.

I think it would be possible to argue that "existing methods" would in fact fail to "break" the circuit-schematic proposal. That is, I think one can make a technically sound case that the circuit schematic proposal meets the "computational infeasibility" requirements of the claims. Of course, the system may turn out to be eventually breakable, but at the the moment I know of no "existing method or equipment" that can solve such a system quickly. While there may be many reasons why the circuit-schematic approach may be too awkward to use in practice, I know of no reason why it should be considered other than "computationally infeasible".

Returning to the nature of their definition of "computational infeasibility", it would seem perhaps desirable to modify the definition to refer to "any computational methods and equipment" rather than to just "existing computational methods and equipment". However, this definition would certainly take us beyond the current state of the art. There are no techniques available for demonstrating that a cryptographic problem is unsolvable by "any" methods. This would require

fundamental breakthroughs in the theory of computer science. There would be no way Stanford could claim to have demonstrated a system that was unsolvable by "any" methods.

There are numerous other technical problems with their definition of "computational infeasibility". For example, it fails to make clear whether it refers to "worst-case" solution time or merely "average" solution time. Also, it gives a particular finite bound (10^{30}) to compare against, whereas the claims do not specify any particular size of knapsack problem to use (although the specifications do make some recommendations).

In summary, I would argue that they have failed to distinguish the inventiveness of their claims 1-6 over the prior art as represented in "Multiuser...". The basis of that argument (the notion of "computational infeasibility") is seriously defective, and the argument itself is inadequate.

On another tack -- is it possible that the publication of the RSA system might invalidate the Stanford claims 1-6? My reading of the actions of the Patent Office would indicate that it thinks that the "Multiuser..." paper does not represent a patentable invention because it "did not teach" a practioner how to effectively implement a public-key cryptosystem. (As noted above, this position requires further argument vis a vis the circuit schematic and other proposals of that paper.) If so, then the general claims to public-key cryptography might then appear to be "up for grabs" for the first person to actually provide an effective implementation of the public-key concept. What evidence do we have that the knapsack systems were invented before RSA was invented or RSA was published? Suppose RSA was first (and I believe it was); would this have any effect on the Stanford claims?

Notes on the Validity of Stanford Patent 4,218,582

Ronald L. Rivest August 10, 1983

There is perhaps a possibility that some "publication" of their work -- different than the "Multiuser..." paper -- is waiting to be uncovered that would invalidate their claims by being more than a year in advance of filing. One possibility is that Diffie and/or Hellman and/or Merkle "spilled the beans" earlier with a conference presentation. The footnote to "New Directions in Cryptography" (by Diffie and Hellman) notes that "portions of the work were presented at:

* The IEEE Information Theory Workshop, Lenox, Mass (June 23-25, 1975)
* The IEEE International Symposium on Information Theory, Ronneby, Sweden, (June 21-24, 1976)."

There is some potential that the latter conference contained very relevant material. I doubt much significant was covered at the earlier one. It would be nice if one of the "knapsack" systems were explictly covered in Sweden. I understand that it would be important if handouts were distributed, and less important otherwise.

This concludes the discussion of the possibility that their claims are invalid, due to lack of inventiveness over prior art.

(2) The Stanford claims 1-6 are overbroad.

There is a tremendous disparity in coverage between claims 1-6 and the remaining claims. Claims 1-6 are very broad and general, covering all the notions of public-key cryptography as disclosed in "Multiuser...". Claims 7-17 are very specific, and only claim a very narrow set of knapsack-based public-key cryptosystems. (Other knapsack-based cryptosystems have been more recently proprosed, which are different than B, B', or L.) Given that claims 1-6 could have been written directly from an examination of the "Multiuser" paper, it seems their validity must stem from a consideration of claims 7-17. Since it is not clear that any usable cryptographic system will result from claims 7-17 (given all the recent research

on breaking knapsack-type systems), it seems unclear why claims 1-6 should be read in a general manner, instead of being read to apply just to the knapsack-type public-key systems. Otherwise, a valuable patent is potentially being granted on the basis of a bluff -- Stanford is claiming rights to all public-key systems for having proposed a system that was supposed to satisfy the requirements of a public-key system but which in the end turns out under closer examination to be easily breakable and totally unusable as a public-key system. It would be similar to being granted the rights to any kind of fusion-based electric power generation on the basis of having demonstrated a system which, while it could generate miniscule amounts of power using fusion, was intrinsically limited and non-expandable. Given that the general idea of electric power generation by nuclear fusion was already published (just as the "Multiuser..." paper represents prior publication for public-key cryptosystems), there should be no basis for someone to claim all methods of fusion-based electric power generation on the basis of having demonstrated one (probably unusable) approach. For these reasons I believe that Stanford claims 1-6 can not be interpreted to cover more than knapsack-based public-key cryptosystems.

(3) The inventor list is improper.

It is my belief that Whit Diffie should be listed as a co-inventor of claims 1-6, if they are to be read broadly. He is, of course, the first author of the "Multiuser..." paper. The fact that he is not listed as an inventor could perhaps be taken as further evidence in support of the contention that claims 1-6 should only be interpreted to cover knapsack-based systems. It would be interesting to hear Diffie's opinion of this.

This concludes my discussion of the weaknesses in Stanford's patent 4,218,582.

B

DISCLOSURE, INCORPORATED
EDGAR DOCUMENT PRINT SUMMARY

DATE PRINTED:	07/25/96
TIME PRINTED:	06:06 P.M.
COMPANY NAME:	SECURITY DYNAMICS TECHNOLOGIES INC
COMPANY NUMBER:	S221700000
DOCUMENT CONTROL#:	96588880
DOCUMENT TYPE:	S-4
DOCUMENT DATE:	06/28/96
AMENDMENT:	
PORTION(S) PRINTED:	1
PAGES PRINTED:	447
CIK#:	0000932064
SEC RECEIPT DATE:	06/28/96
SEC FILE#:	33307265

AS FILED WITH THE SECURITIES AND EXCHANGE COMMISSION ON JUNE 28, 1996
REGISTRATION NO. 333-

SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, D.C. 20549

FORM S-4

REGISTRATION STATEMENT
UNDER
THE SECURITIES ACT OF 1933

SECURITY DYNAMICS TECHNOLOGIES, INC.
(Exact name of registrant as specified in its charter)

DELAWARE	3577	04-2916506
(State or other jurisdiction of incorporation or organization)	(Primary Standard Industrial Classification Code Number)	(I.R.S. Employer Identification Number)

ONE ALEWIFE CENTER
CAMBRIDGE, MASSACHUSETTS 02140
(617) 547-7820
(Address, including zip code, and telephone number, including area code, of registrant's principal executive offices)

CHARLES R. STUCKEY, JR.
PRESIDENT AND CHIEF EXECUTIVE OFFICER
SECURITY DYNAMICS TECHNOLOGIES, INC.
ONE ALEWIFE CENTER
CAMBRIDGE, MASSACHUSETTS 02140
(617) 547-7820
(Name, address, including zip code, and telephone number, including area code, of agent for service)

WITH COPIES TO:

HAL J. LEIBOWITZ, ESQ.
JAY E. BOTHWICK, ESQ.
HALE AND DORR
60 STATE STREET
BOSTON, MASSACHUSETTS 02109
(617) 526-6000

DAVID W. HEALY, ESQ.
ROBERT B. DELLENBACH, ESQ.
TRAM T. PHI, ESQ.
JOHN F. PLATZ, ESQ.
FENWICK & WEST LLP
TWO PALO ALTO SQUARE
PALO ALTO, CALIFORNIA 94306
(415) 494-0600

APPROXIMATE DATE OF COMMENCEMENT OF PROPOSED SALE OF SECURITIES TO THE PUBLIC: As soon as practicable after the effective date of this Registration Statement and the satisfaction or waiver of certain other conditions under the Agreement and Plan of Merger described herein.

INTERESTS OF CERTAIN PERSONS IN THE MERGER

As of June 3, 1996, directors and executive officers of SDI and their affiliates may be deemed to be beneficial owners of approximately 5.7% of the outstanding shares of SDI Common Stock. See "Management -- SDI -- Security Ownership of Certain Beneficial Owners and Management" for additional information concerning such ownership. Each of the directors and executive officers of SDI has advised SDI that he or she intends to vote or direct the vote of all the outstanding shares of SDI Common Stock over which he or she has voting control in favor of approval of the Merger Proposal.

As of June 28, 1996, directors of RSA and their affiliates may be deemed to be beneficial owners of approximately 73% of the outstanding shares of RSA Stock. See "Management -- RSA -- Security Ownership of Certain Beneficial Owners and Management" for additional information concerning such ownership. As discussed below under "The Merger Agreement -- Related Agreements -- Stockholder Agreements," each of the directors of RSA has agreed to vote or direct the vote of all of the outstanding shares of RSA Stock over which he has voting control in favor of the approval of the Merger Agreement and the Merger.

Simultaneously with the execution of the Merger Agreement, SDI, RSA and D. James Bidzos entered into an Employment Agreement pursuant to which Mr. Bidzos will serve as the President of the Surviving Corporation for a period of two years following the Effective Time, subject to earlier termination in accordance with the terms of such Agreement. In addition, a condition to SDI's obligations to consummate the Merger pursuant to the Merger Agreement is the execution of three-year employment agreements with RSA by C. Victor Chang, Burton Kaliski and Paul Livesay, employees of RSA. See "The Merger Agreement -- Related Agreements -- Other Employment Agreements."

It is currently anticipated that, following the Effective Time, the SDI Board will vote to expand the size of the SDI Board by one and to elect Mr. Bidzos to the SDI Board.

Pursuant to the Stockholder Agreements, SDI has agreed to provide Messrs. Bidzos, Fischer and Rivest, the directors of RSA, with certain rights with respect to the registration under the Securities Act of the shares of SDI Common Stock to be held by Messrs. Bidzos, Fischer and Rivest following the Effective Time of the Merger, including the right in certain circumstances to require SDI to prepare and file registration statements under the Securities Act with respect to such shares and the right to include such shares in any registration conducted by SDI, subject to certain cutbacks. See "The Merger Agreement -- Related Agreements -- Stockholder Agreements."

In addition, pursuant to the Stockholder Agreements, each of Messrs. Bidzos, Fischer and Rivest has appointed each of the directors of SDI as such stockholder's proxy during the period specified in the Stockholder Agreements to vote all RSA Stock then owned by such stockholder in favor of approval of the Merger and the Merger Agreement and against approval of any proposal made in opposition to or in competition with the consummation of the Merger and the Merger Agreement, including any merger, consolidation, sale of assets, reorganization or recapitalization of RSA with any party other than SDI and against any liquidation or winding up of RSA. See "The Merger Agreement -- Related Agreements -- Stockholder Agreements."

Ms. Conrow, the Chief Financial Officer of RSA, holds an option to acquire shares of RSA Common Stock subject to repurchase rights in favor of RSA, which repurchase rights will terminate with respect to 10,000 shares upon the closing of the Merger pursuant to the terms of an employment agreement between RSA and Ms. Conrow.

ACCOUNTING TREATMENT

The Merger is intended to qualify as a pooling of interests for accounting

C

United States Patent [19]

Hellman et al.

[11] **4,218,582**

[45] **Aug. 19, 1980**

[54] **PUBLIC KEY CRYPTOGRAPHIC APPARATUS AND METHOD**

[75] Inventors: **Martin E. Hellman**, Stanford; **Ralph C. Merkle**, Palo Alto, both of Calif.

[73] Assignee: **The Board of Trustees of the Leland Stanford Junior University**, Stanford, Calif.

[21] Appl. No.: **839,939**

[22] Filed: **Oct. 6, 1977**

[51] **Int. Cl.**[2] **H04L 9/04**

[52] **U.S. Cl.** **178/22**; 364/900

[58] **Field of Search** 178/22

[56] **References Cited**

PUBLICATIONS

"New Directions in Cryptography," Diffie et al., *IEEE Transactions on Information Theory*, vol. II22, No. 6, Nov. 1976, pp. 644–654.

"A User Authentication Scheme not Requiring Secrecy in the Computer," Evans, Jr., et al., *Communications of the ACM*, Aug. 1974, vol. 17, No. 8, pp. 437–442.

"A High Security Log-In Procedure," Purdy, *Communications of the ACM*, Aug. 1974, vol. 17, No. 8, pp. 442–445.

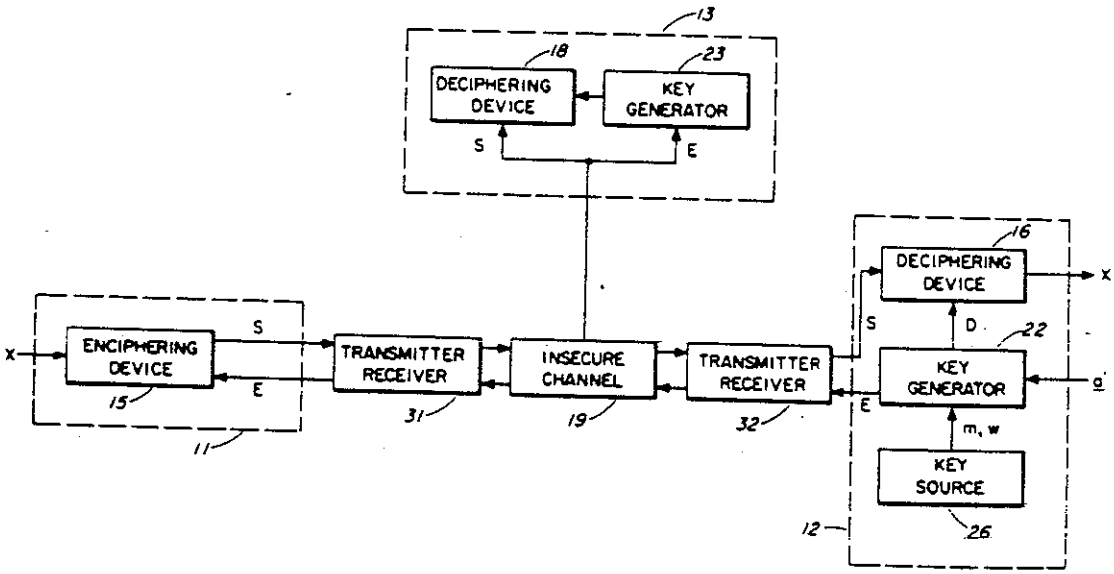
Diffie et al., "Multi-User Cryptographic Techniques," *AFIPS Conference Proceedings*, vol. 45, pp. 109–112, Jun. 8, 1976.

Primary Examiner—Howard A. Birmiel

[57] **ABSTRACT**

A cryptographic system transmits a computationally secure cryptogram that is generated from a publicly known transformation of the message sent by the transmitter; the cryptogram is again transformed by the authorized receiver using a secret reciprocal transformation to reproduce the message sent. The authorized receiver's transformation is known only by the authorized receiver and is used to generate the transmitter's transformation that is made publicly known. The publicly known transformation uses operations that are easily performed but extremely difficult to invert. It is infeasible for an unauthorized receiver to invert the publicly known transformation or duplicate the authorized receiver's secret transformation to obtain the message sent.

17 Claims, 13 Drawing Figures



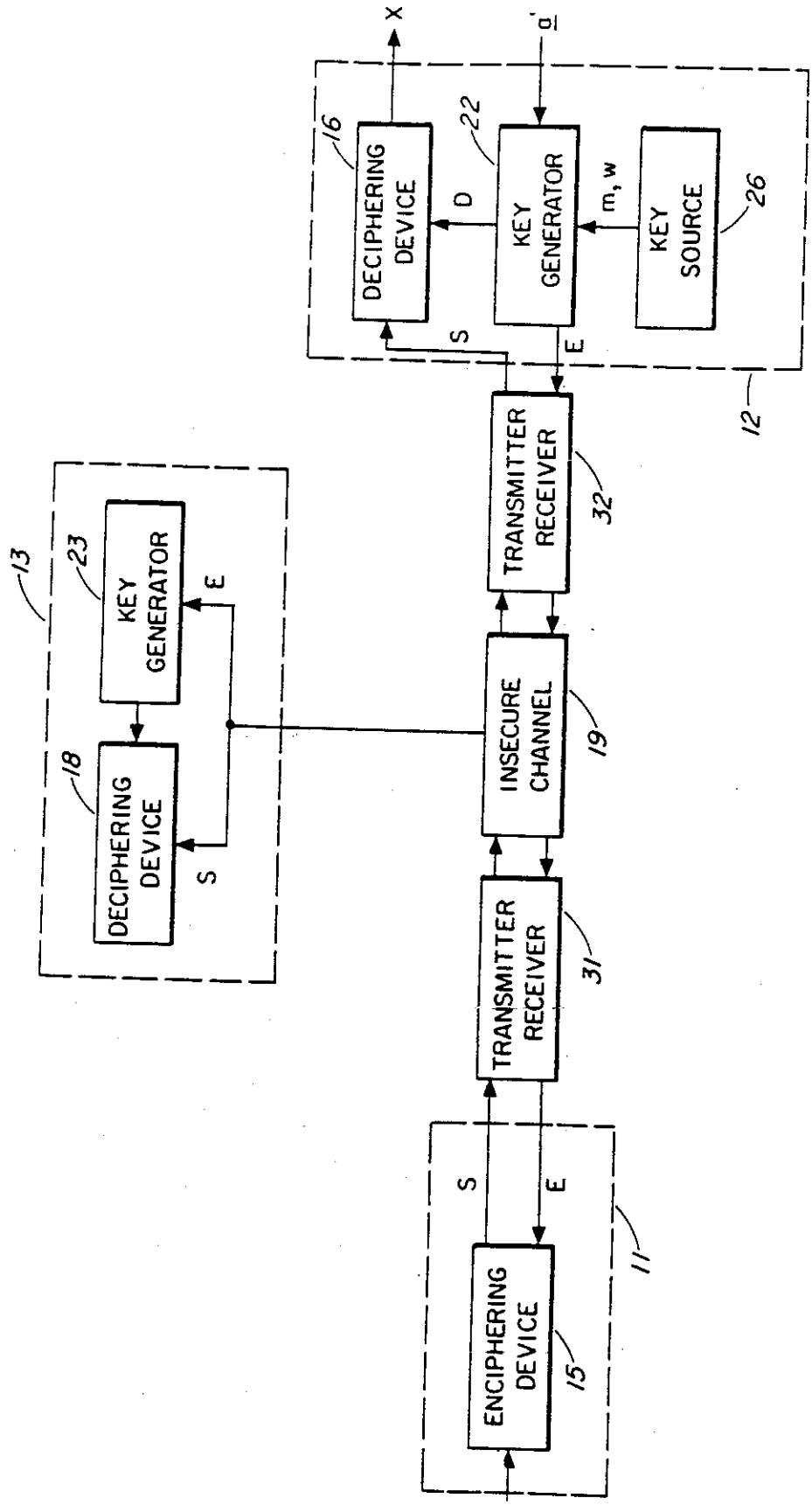

X
a
DECIPHERING DEVICE
16
D
KEY GENERATOR
22
m, w
KEY SOURCE
26
S
E
12
TRANSMITTER RECEIVER
32
13
KEY GENERATOR
23
E
INSECURE CHANNEL
19
DECIPHERING DEVICE
18
S
TRANSMITTER RECEIVER
31
S
E
ENCIPHERING DEVICE
15
11

FIG. 1

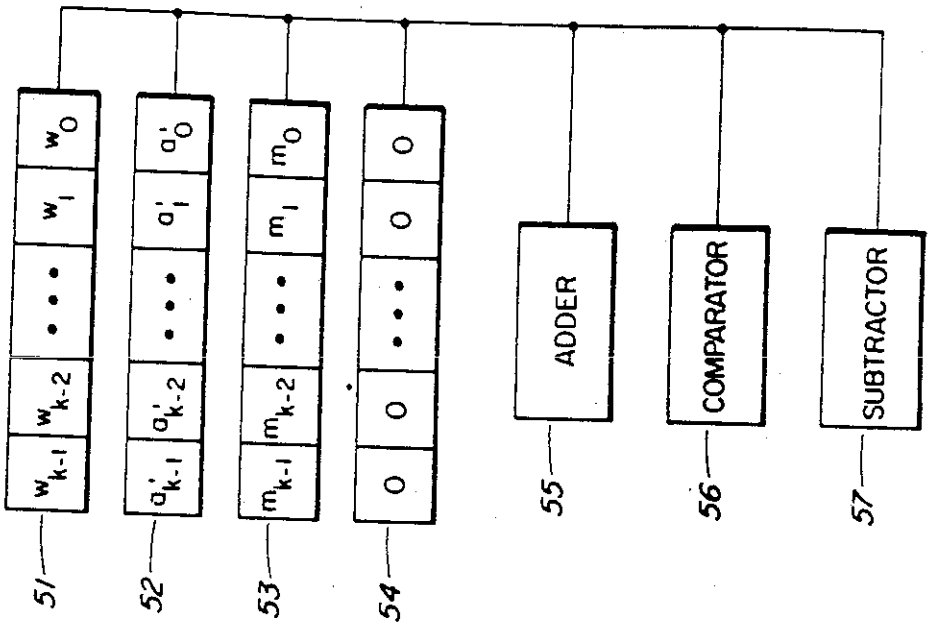

w_0 w_1 w_{k-2} w_{k-1}
a'_0 a'_1 a'_{k-2} a'_{k-1}
m_0 m_1 m_{k-2} m_{k-1}
0 0 0 0
ADDER
COMPARATOR
SUBTRACTOR
51
52
53
54
55
56
57

FIG. 3

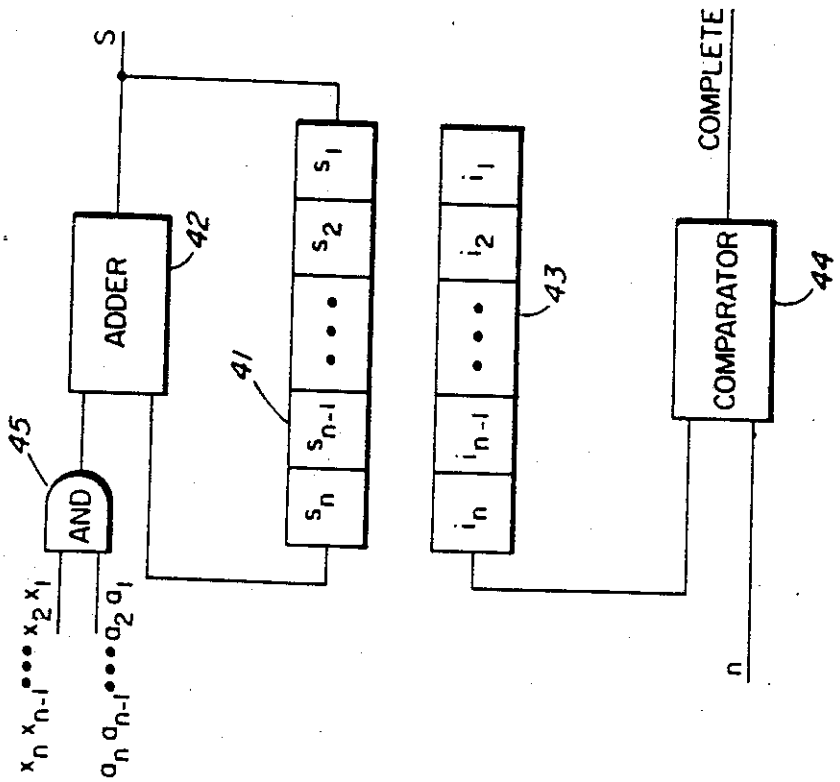

S
ADDER
42
41
s_1 s_2 s_{n-1} s_n
i_1 i_2 i_{n-1} i_n
43
COMPLETE
COMPARATOR
44
45
AND
$x_n x_{n-1} \cdots x_2 x_1$
$a_n a_{n-1} \cdots a_2 a_1$
n

FIG. 2

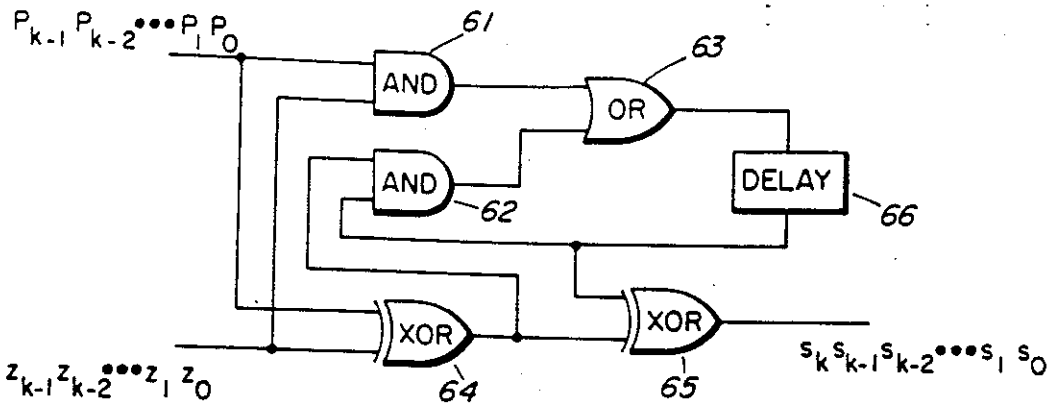


FIG. 4

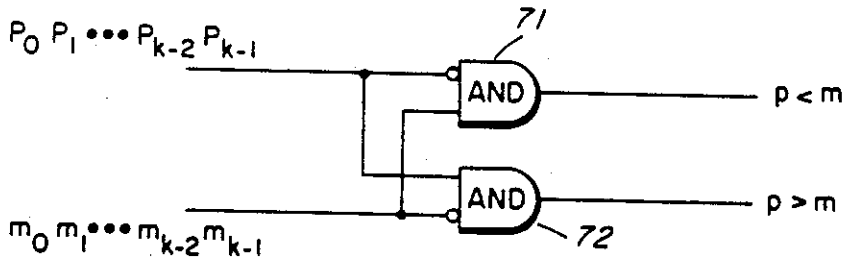


FIG. 5

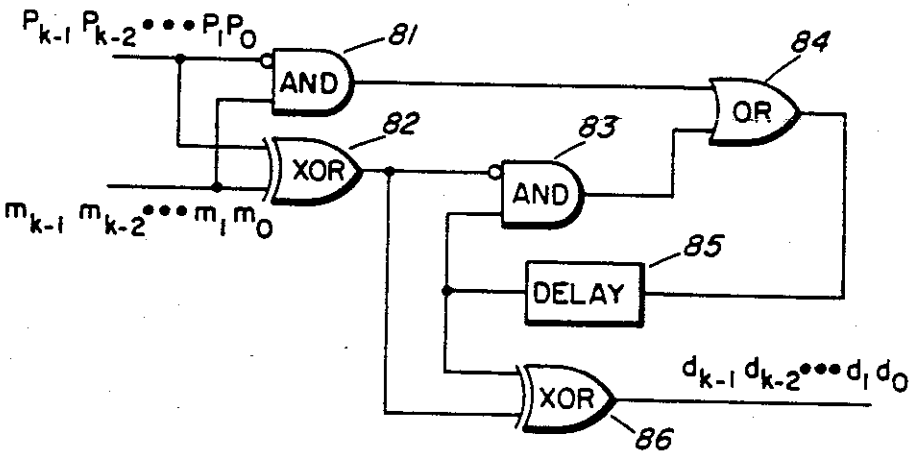


FIG. 6

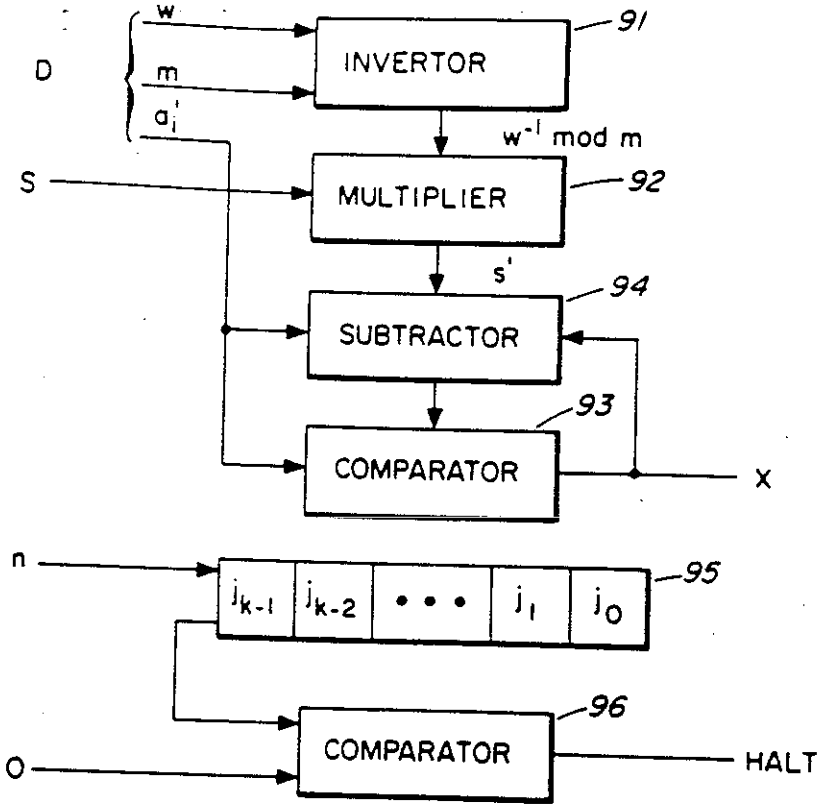

D
w
m
a_i'
INVERTOR
91
w^{-1} mod m
S
MULTIPLIER
92
s'
94
SUBTRACTOR
93
COMPARATOR
X
n
j_{k-1} j_{k-2} • • • j_1 j_0
95
96
COMPARATOR
0
HALT

FIG. 7

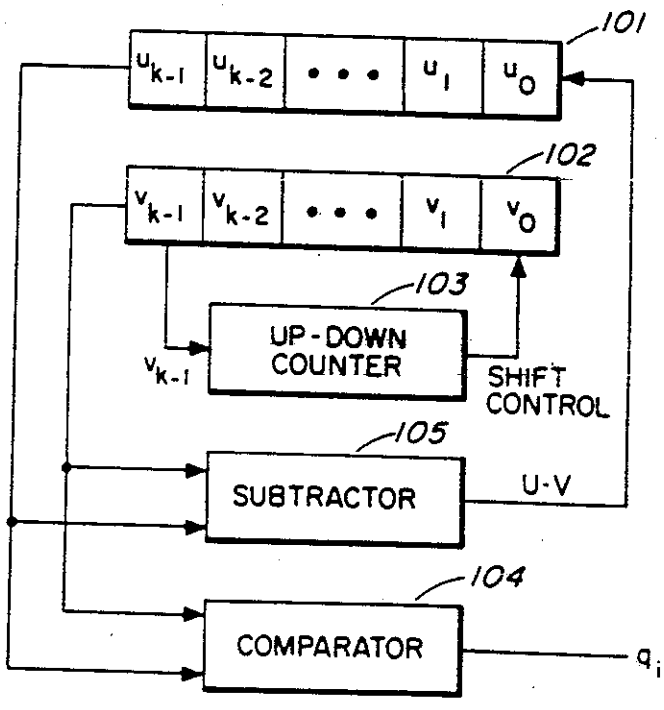

101
u_{k-1} u_{k-2} • • • u_1 u_0
102
v_{k-1} v_{k-2} • • • v_1 v_0
103
UP-DOWN COUNTER
v_{k-1}
SHIFT CONTROL
105
SUBTRACTOR
U-V
104
COMPARATOR
q_i

FIG. 8

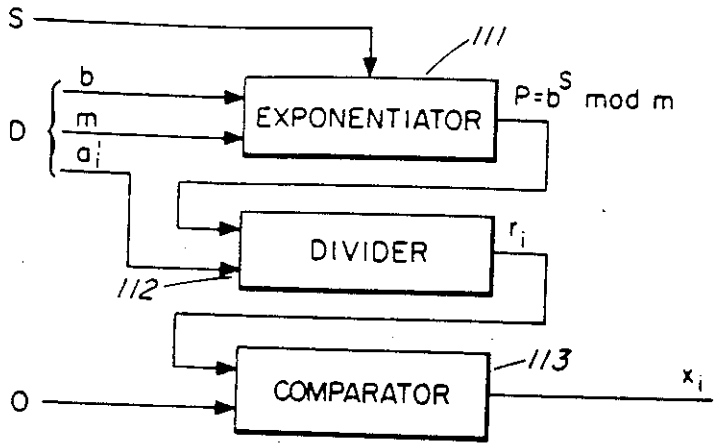

S
111
b
m
D
a_i
EXPONENTIATOR
$P=b^S \bmod m$
DIVIDER
r_i
112
113
COMPARATOR
x_i
0

FIG. 9

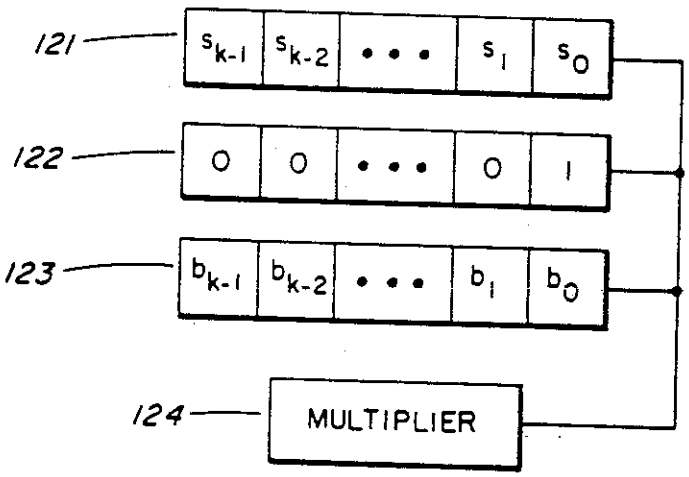

121
s_{k-1} s_{k-2} • • • s_1 s_0
122
0 0 • • • 0 1
123
b_{k-1} b_{k-2} • • • b_1 b_0
124
MULTIPLIER

FIG. 10

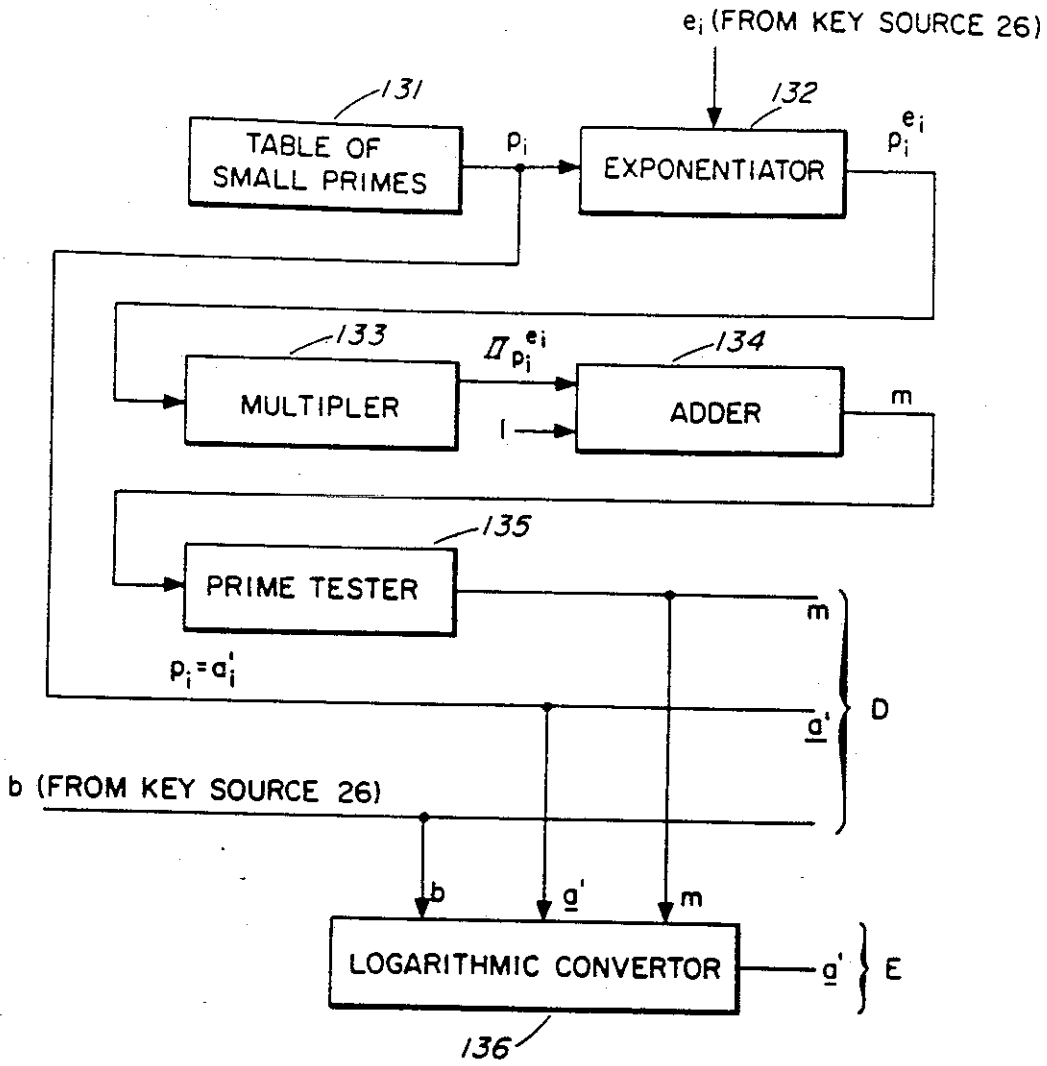

e_i (FROM KEY SOURCE 26)
131
TABLE OF SMALL PRIMES
p_i
132
EXPONENTIATOR
$p_i^{e_i}$
133
MULTIPLER
$\Pi p_i^{e_i}$
1
134
ADDER
m
135
PRIME TESTER
m
$p_i = a'_i$
a'
D
b (FROM KEY SOURCE 26)
b
a'
m
LOGARITHMIC CONVERTOR
a'
E
136

FIG. 11

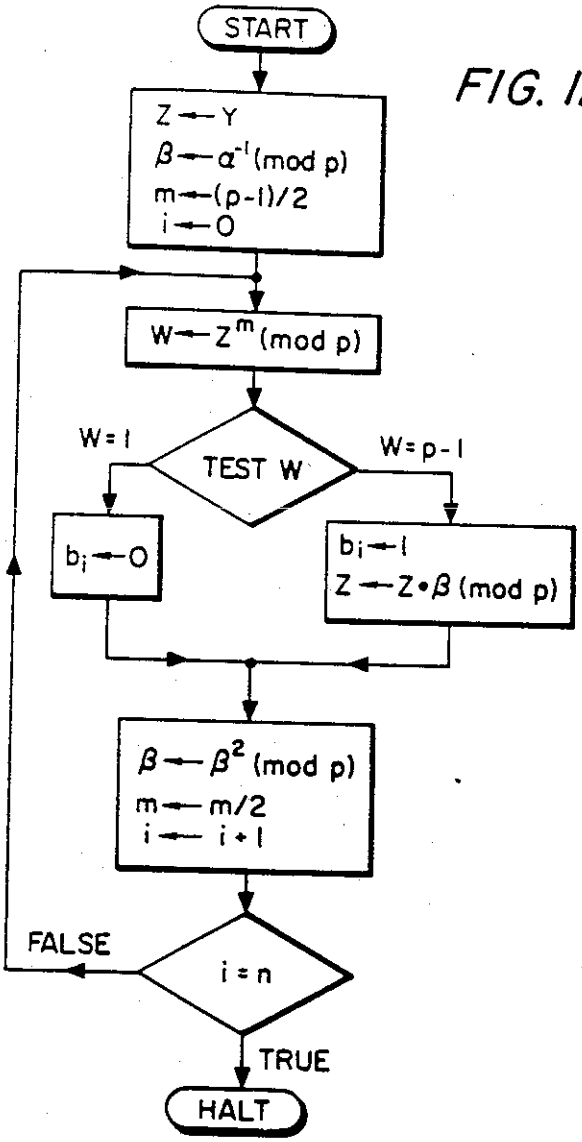

START
Z ← Y
β ← α^-1 (mod p)
m ← (p-1)/2
i ← 0
W ← Z^m (mod p)
W = 1
TEST W
W = p-1
b_i ← 0
b_i ← 1
Z ← Z•β (mod p)
β ← β^2 (mod p)
m ← m/2
i ← i+1
FALSE
i = n
TRUE
HALT

FIG. 12

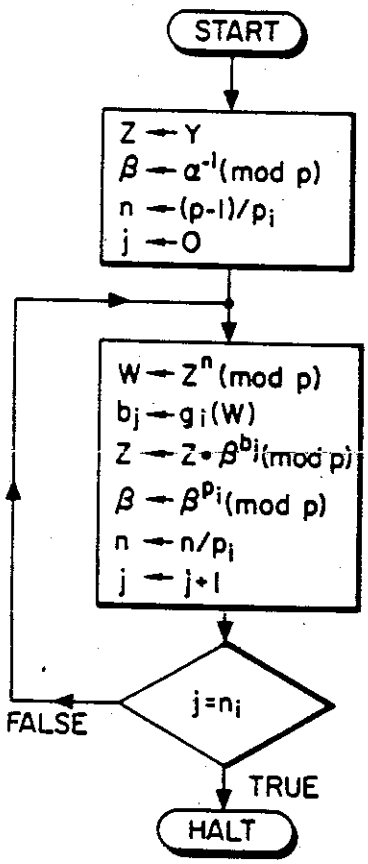

START
Z ← Y
β ← α^-1 (mod p)
n ← (p-1)/p_i
j ← 0
W ← Z^n (mod p)
b_j ← g_i(W)
Z ← Z•β^b_j (mod p)
β ← β^p_i (mod p)
n ← n/p_i
j ← j+1
j = n_i
FALSE
TRUE
HALT

FIG. 13

PUBLIC KEY CRYPTOGRAPHIC APPARATUS AND METHOD

The Government has rights in this invention pursuant to Grant No. ENG-10173 of the National Science Foundation and IPA No. 0005.

BACKGROUND OF THE INVENTION

1. Field of Invention

The invention relates to cryptographic systems.

2. Description of Prior Art

Cryptographic systems are widely used to ensure the privacy and authenticity of messages communicated over insecure channels. A privacy system prevents the extraction of information by unauthorized parties from messages transmitted over an insecure channel, thus assuring the sender of a message that it is being read only by the intended receiver. An authentication system prevents the unauthorized injection of messages into an insecure channel, assuring the receiver of the message of the legitimacy of its sender.

Currently, most message authentication consists of appending an authenticator pattern, known only to the transmitter and intended receiver, to each message and encrypting the combination. This protects against an eavesdropper being able to forge new, properly authenticated messages unless he has also stolen the cipher key being used. However, there is little protection against the threat of dispute; that is, the transmitter may transmit a properly authenticated message, later deny this action, and falsely blame the receiver for taking unauthorized action. Or, conversely, the receiver may take unauthorized action, forge a message to itself, and falsely blame the transmitter for these actions. The threat of dispute arises out of the absence of a suitable receipt mechanism that could prove a particular message was sent to a receiver by a particular transmitter.

One of the principal difficulties with existing cryptographic systems is the need for the sender and receiver to exchange a cipher key over a secure channel to which the unauthorized party does not have access. The exchange of a cipher key frequently is done by sending the key in advance over a secure channel such as private courier or registered mail; such secure channels are usually slow and expensive.

Diffie, et al, in "Multiuser Cryptographic Techniques," *AFIPS-Conference Proceedings*, Vol. 45, pp. 109–112, June 8, 1976, propose the concept of a public key cryptosystem that would eliminate the need for a secure channel by making the sender's keying information public. It is also proposed how such a public key cryptosystem could allow an authentication system which generates an unforgeable message dependent digital signature. Diffie presents the idea of using a pair of keys E and D, for enciphering and deciphering a message, such that E is public information while D is kept secret by the intended receiver. Further, although D is determined by E, it is infeasible to compute D from E. Diffie suggests the plausibility of designing such a public key cryptosystem that would allow a user to encipher a message and send it to the intended receiver, but only the intended receiver could decipher it. While suggesting the plausibility of designing such systems, Diffie presents neither proof that public key cryptosystems exist, nor a demonstration system.

Diffie suggests three plausibility arguments for the existence of a public key cryptosystem: a matrix approach, a machine language approach and a logic mapping approach. While the matrix approach can be designed with matrices that require a demonstrably infeasible cryptanalytic time (i.e., computing D from E) using known methods, the matrix approach exhibits a lack of practical utility because of the enormous dimensions of the required matrices. The machine language approach and logic mapping approach are also suggested, but there is no way shown to design them in such a manner that they would require demonstrably infeasible cryptanalytic time.

Diffie also introduces a procedure using the proposed public key cryptosystems, that could allow the receiver to easily verify the authenticity of a message, but which prevents him from generating apparently authenticated messages. Diffie describes a protocol to be followed to obtain authentication with the proposed public key cryptosystem. However, the authentication procedure relies on the existence of a public key cryptosystem which Diffie did not provide.

SUMMARY AND OBJECTS OF THE INVENTION

Accordingly, it is an object of the invention to allow authorized parties to a conversation (conversers) to converse privately even though an unauthorized party (eavesdropper) intercepts all of their communications.

Another object of this invention is to allow a converser on an insecure channel to authenticate another converser's identity.

Another object of this invention is to provide a receipt to a receiver on an insecure channel to prove that a particular message was sent to the receiver by a particular transmitter. The object being to allow the receiver to easily verify the authenticity of a message, but to prevent the receiver from generating apparently authenticated messages.

An illustrated embodiment of the present invention describes a method and apparatus for communicating securely over an insecure channel, by communicating a computationally secure cryptogram that is a publicly known transformation of the message sent by the transmitter. The illustrated embodiment differs from prior approaches to a public key cryptosystem, as described in "Multiuser Cryptographic Techniques," in that it is both practical to implement and is demonstrably infeasible to invert using known methods.

In the present invention, a receiver generates a secret deciphering key and a public enciphering key, such that the secret deciphering key is difficult to generate from the public enciphering key. The transmitter enciphers a message to be communicated by transforming the message with the public enciphering key, wherein the transformation used to encipher the message is easy to effect but difficult to invert without the secret deciphering key. The enciphered message is then communicated from the transmitter to the receiver. The receiver deciphers the enciphered message by inverting the enciphering transformation with the secret deciphering key.

Another illustrated embodiment of the present invention describes a method and apparatus for allowing a transmitter to authenticate an authorized receiver's identity. The authorized receiver generates a secret deciphering key and a public enciphering key, such that the secret deciphering key is difficult to generate from the public enciphering key. The transmitter enciphers a message to be communicated by transforming the message with the public enciphering key, wherein the trans-

formation used to encipher the message is easy to effect but difficult to invert without the secret deciphering key. The enciphered message is then transmitted from the transmitter to the receiver. The receiver deciphers the enciphered message by inverting the enciphering transformation with the secret deciphering key. The receiver's identity is authenticated to the transmitter by the receiver's ability to decipher the enciphered message.

Another illustrated embodiment of the present invention describes a method and apparatus for providing a receipt for a communicated message. A transmitter generates a secret key and a public key, such that the secret key is difficult to generate from the public key. The transmitter then generates a receipt by transforming a representation of the communicated message with the secret key, wherein the transformation used to generate the receipt is difficult to effect without the secret key and easy to invert with the public key. The receipt is then communicated from the transmitter to the receiver. The receiver inverts the transformation with the public key to obtain the representation of the communicated message from the receipt and validates the receipt by comparing the similarity of the representation of the communicated message with the communicated message.

Additional objects and features of the present invention will appear from the description that follows wherein the preferred embodiments have been set forth in detail in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a public key cryptosystem that transmits a computationally secure cryptogram over an insecure communication channel.

FIG. 2 is a block diagram of an enciphering device for enciphering a message into ciphertext in the public key cryptosystem of FIG. 1.

FIG. 3 is a block diagram of a multiplier for performing modular multiplications in the deciphering device of FIG. 7, the exponentiator of FIG. 10, and the public key generator of FIG. 11.

FIG. 4 is a detailed schematic diagram of an adder for performing additions in the enciphering device of FIG. 2, the multiplier of FIG. 3, and the public key generator of FIG. 11.

FIG. 5 is a detailed schematic diagram of a comparator for performing magnitude comparisons in the enciphering device of FIG. 2, the multiplier of FIG. 3, the deciphering device of FIG. 7, the divider of FIG. 8, and the alternative deciphering device of FIG. 9.

FIG. 6 is a detailed schematic diagram of a subtractor for performing subtraction in the multiplier of FIG. 3, the deciphering device of FIG. 7, and the dividier of FIG. 8.

FIG. 7 is a block diagram of a deciphering device for deciphering a ciphertext into message in the public key cryptosystem of FIG. 1.

FIG. 8 is a block diagram of a divider for performing division in the invertor of FIG. 7 and the alternative deciphering device of FIG. 9.

FIG. 9 is a block diagram of an alternative deciphering device for deciphering a ciphertext into message in the public key cryptosystem of FIG. 1.

FIG. 10 is an exponentiator for raising various numbers to various powers in modulo arithmetic in the alternative deciphering device of FIG. 9 and the public key generator of FIG. 11.

FIG. 11 is a public key generator for generating the public enciphering key in the public key cryptosystem of FIG. 1.

FIG. 12 is a flow chart for the algorithm of the logarithmic converter of FIG. 11 when $p-1$ is a power of 2.

FIG. 13 is a flow chart for the algorithm for computing the coefficients $\{b_j\}$ of the expansion

$$x(\mathrm{mod}\ p_i^{n_i}) = \sum_{j=0}^{n_i-1} b_j p^j$$

where $0 \leqq b_j \leqq p_i - 1$, of the logarithmic convertor of FIG. 11, when $p-1$ is not a power of 2.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 1, a public key cryptosystem is shown in which all transmissions take place over an insecure communication channel 19, for example a telephone line. Communication is effected on the insecure channel 19 between transmitter 11 and receiver 12 using transmitter-receiver units 31 and 32, which may be modems such as Bell 201 modems. Transmitter 11 possesses an unenciphered or plaintext message X to be communicated to receiver 12. Transmitter 11 and receiver 12 include an enciphering device 15 and deciphering device 16 respectively, for enciphering and deciphering information under the action of an enciphering key E on line E and a reciprocal deciphering key D on line D. The enciphering and deciphering devices 15 and 16 implement inverse transformations when loaded with the corresponding keys E and D. For example, the keys may be a sequence of random letters or digits. The enciphering device 15 enciphers the plaintext message X into an enciphered message or ciphertext S that is transmitted by transmitter 11 through the insecure channel 19; the ciphertext S is received by receiver 12 and deciphered by deciphering device 16 to obtain the plaintext message X. An unauthorized party or eavesdropper 13 is assumed to have key generator 23 and deciphering device 18 and to have access to the insecure channel 19, so if he knew the deciphering key D he could decipher the ciphertext S to obtain the plaintext message X.

The example system makes use of the difficulty of the so-called "knapsack problem." Definitions of the knapsack problem exist in the literature, for example, Ellis Horowitz and Sartaj Sahni, "Computing Partitions with Applications to the Knapsack Problem", *JACM*, Vol. 21, No. 2, April 1974, pp. 277–292; and O. H. Ibarra and C. E. Kim, "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems", *JACM*, Vol. 22, No. 4, October 1975, pp. 464–468. The definition used here is adapted from R. M. Karp, "Reducibility Among Combinatorial Problems" in Complexity of Computer Computations, by R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York (1972), pp. 85–104. Simply stated, given a one-dimensional knapsack of length S and a vector a composed of n rods of lengths $a_1, a_2, \ldots a_n$, the knapsack problem is to find a subset of the rods which exactly fills the knapsack, if such a subset exists. Equivalently, find a binary n-vector x of 0's and 1's such that $S = a^*x$, if such an x exists, (* applied to vectors denotes dot product, applied to scalars denotes normal multiplication).

A supposed solution, x, is easily checked in at most n additions; but, to the best of current knowledge, finding a solution requires a number of operations which grows exponentially in n. Exhaustive trial and error search over all 2^n possible x's is computationally infeasible if n is larger than one or two hundred. Thus, it is computationally infeasible to invert the transformation; such transformations are characterized by the class of mathematical functions known as one-way cipher functions. A task is considered computationally infeasible if its cost as measured by either the amount of memory used or the computing time is finite but impossibly large, for example, on the order of approximately 10^{30} operations with existing computational methods and equipment.

Theory suggests the difficulty of the knapsack problem because it is an NP-complete problem, and is therefore one of the most difficult computational problems of a cryptographic nature. (See for example, A. V. Aho, J. E. Hopcraft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading, Ma.; Addison-Wesley, 1974, pp. 363–404.) Its degree of difficulty, however, is dependent on the choice of a. If $a=(1, 2, 4, \ldots 2^{(n-1)})$, then solving for x is equivalent to finding the binary representation of S. Somewhat less trivially, if for all i,

$$a_i > \sum_{j=1}^{i-1} a_j \tag{1}$$

then x is also easily found: $x_n=1$ if and only if $S \geqq a_n$, and, for $i=n-1, n-2, \ldots 1$, $x_i=1$ if and only if

$$S - \sum_{j=i+1}^{n} x_j * a_j \geqq a_i \tag{2}$$

If the components of x are allowed to take on integer values between 0 and l then condition (1) can be replaced by

$$a_i > l \sum_{j=1}^{i-1} a_j$$

and x_i can be recovered as the integer part of

$$(S - \sum_{j=i+1}^{n} x_j * a_j)/a_i.$$

Equation (2) for evaluating x_i when x_i is binary valued is equivalent to this rule for l=1.

A trap door knapsack is one in which careful choice of a allows the designer to easily solve for any x, but which prevents anyone else from finding the solution. Two methods will be described for constructing trap door knapsacks, but first a description of their use in a public key cryptosystem as shown in FIG. 1 is provided. Receiver 12 generates a trap door knapsack vector a, and either places it in a public file or transmits it to transmitter 11 over the insecure channel 19. Transmitter 11 represents the plaintext message X as a vector x of n 0's and 1's, computes $S=a*x$, and transmits S to receiver 12 over the insecure channel 19. Receiver 12 can solve S for x, but it is infeasible for eavesdropper 13 to solve S for x.

In one method for generating trap door knapsacks, the key generator 22, uses random numbers generated by key source 26 to select two large integers, m and w, such that w is invertible modulo m, (i.e., so that m and w have no common factors except 1). For example, the key source 26 may contain a random number generator that is implemented from noisy amplifiers (e.g., Fairchild μ 709 operational amplifiers) with a polarity detector. The key generator 22 is provided a knapsack vector, a' which satisfies (1) and therefore allows solution of $S'=a'*x$, and transforms the easily solved knapsack vector a' into a trap door knapsack vector a via the relation

$$a_i = w * a'_i \bmod m \tag{3}$$

The vector a serves as the public enciphering key E on line E, and is either placed in a public file or transmitted over the insecure channel 19 to transmitter 11. The enciphering key E is thereby made available to both the transmitter 11 and the eavesdropper 13. The transmitter 11 uses the enciphering key E, equal to a, to generate the ciphertext S from the plaintext message X, represented by vector x, by letting $S=a*x$. However, because the a_i may be psuedo-randomly distributed, the eavesdropper 13 who knows a, but not w or m, cannot feasibly solve a knapsack problem involving a to obtain the desired message x.

The deciphering device 16 of receiver 12 is given w, m and a' as its secret deciphering key D, and can easily compute

$$\begin{aligned} S' &= 1/w * S \bmod m && (4)\\ &= 1/w * \Sigma x_i * a_i \bmod m && (5)\\ &= 1/w * \Sigma x_i * w * a'_i \bmod m && (6)\\ &= \Sigma x_i * a'_i \bmod m && (7) \end{aligned}$$

If m is chosen so that

$$m > \Sigma a_i' \tag{8}$$

then (7) implies that S' is equal to $\Sigma x_i * a_i'$ in integer arithmetic as well as mod m. This knapsack is easily solved for x, which is also the solution to the more difficult trap door knapsack problem $S=a*x$. Receiver 12 is therefore able to recover the plaintext message X, represented as the binary vector x. But, the eavesdropper 13's trap door knapsack problem can be made computationally infeasible to solve, thereby preventing the eavesdropper 13 from recovering the plaintext message X.

To help make these ideas more clear, an illustrative example is given in which $n=5$. Taking $m=8443$, $a'=(171, 196, 457, 1191, 2410)$, and $w=2550$, then $a=(5457, 1663, 216, 6013, 7439)$. Given $x=(0, 1, 0, 1, 1)$ the enciphering device 15 computes $S=1663+6013+7439=15115$. The deciphering device 16 uses Euclid's algorithm (see for instance, D. Knuth, *The Art of Computer Programming*, vol. II, Addison-Wesley, 1969, Reading Ma.) to compute $1/w=3950$ and then computes

$$\begin{aligned} S' &= 1/w * S \bmod m \\ &= 3950 * 15115 \bmod 8443 \\ &= 3797 \end{aligned} \tag{9}$$

Because $S' > a_5'$, the deciphering device 16 determines that $x_5=1$. Then, using (2) for the a' vector, it determines that $x_4=1$, $x_3=0$, $x_2=1$, $x_1=0$ or $x=(0, 1, 0, 1, 1)$, which is also the correct solution to $S=a*x$.

The eavesdropper, 13 who does not know m, w or a' has great difficulty in solving for x in $S=a*x$ even

though he knows the method used for generating the trap door knapsack vector a. His task can be made infeasible by choosing larger values for n, m, w and a'. His task can be further complicated by scrambling the order of the a_i, and by adding different random multiples of m to each of the a_i.

The example given was extremely small in size and only intended to illustrate the technique. Using $n=100$ (which is the lower end of the usable range for high security systems at present) as a more reasonable value, it is suggested that m be chosen approximately uniformly from the numbers between $2^{201}+1$ and $2^{202}-1$; that a_1' be chosen uniformly from the range $(1, 2^{100})$; that a'_2 be chosen uniformly from $(2^{100}+1, 2*2^{100})$; that a_3' be chosen uniformly from $(3\times 2^{100}+1, 4*2^{100})$; . . . and that a_i' be chosen uniformly from $((2^{i-1}-1)*2^{100}+1, 2^{i-1}*2^{100})$; and that w' be chosen uniformly from (2, m−2) and then divided by the greatest common divisor of (w', m) to yield w.

These choices ensure that (8) is met and that an eavesdropper **13** has at least 2^{100} possibilities for each parameter and hence cannot search over them all.

The enciphering device **15** is shown in FIG. **2**. The sequence of integers $a_1, a_2, \ldots a_n$ is presented sequentially in synchronization with the sequential presentation of 0's and 1's of $x_1, x_2, \ldots x_n$. The S register **41** is initially set to zero. If $x_i=1$ the S register **41** contents are a_i are added by adder **42** and the result placed in the S register **41**. If $x_i=0$ the contents of the S register **41** are left unchanged. In either event, i is replaced by i+1 until i=n, in which case the enciphering operation is complete. The i register **43** is initially set to zero and incremented by 1 after each cycle of the enciphering device. Either the adder **42**, or a special up counter can be used to increment the i register **43** contents. With the range of values suggested above, the S and i registers **41** and **43** both can be obtained from a single 1024 bit random access memory (RAM) such as the Intel 2102. The implementation of the adder **42** will be described in more detail later, as will the implementation of a comparator **44** required for comparing i and n to determine when the enciphering operation is complete.

The key generaor **22** comprises a modulo m multiplier, such as that depicted in FIG. **3**, for producing $a_i=w*a_i'$ mod m. The two numbers w and a_i' to be multiplied are loaded into the W and A' registers **51** and **52** respectively, and m is loaded into the M register **53**. The product $w*a_i'$ modulo m will be produced in the P register **54** which is initially set to zero. If k, the number of bits in the binary representation of m, is 200, then all four registers can be obtained from a single 1024 bit RAM such as the Intel 2102. The implementation of FIG. **3** is based on the fact that wa_i' mod $m=w_0a_i'$ mod $m+2\ w_1a_i'$ mod $m+4\ w_2a_i'$ mod $m+\ \ldots\ +2^{k-1}\ w_{k-1}a_i'$ mod m.

To multiply w times a_i', if the rightmost bit, containing w_o of the W register **51** is 1 then the contents of the A' register **53** are added to the P register **54** by adder **55**. If $w_o=0$, then the P register **54** is unchanged. Then the M and P register contents are compared by comparator **56** to determine if the contents of the P register **54** are greater than or equal to m, the contents of the M register **53**. If the contents of the P register **54** are greater than or equal to m then subtractor **57** subtracts m from the contents of the P register **54** and places the difference in the P register **54**, if less than m the P register **54** is unchanged.

Next, the contents of W register **51** are shifted one bit to the right and a 0 is shifted in at the left so its contents become $0w_{k-1}\ w_{k-2}\ \ldots\ w_2w_1$, so that w is ready for conputing $2w_1a'$ mod m. The quantity of 2a' mod m is computed for this purpose by using adder **55** to add a' to itself, using comparator **56** to determine if the result, 2a', is less than m, and using subtractor **57** for subtracting m from 2a' if the result is not less than m. The result, 2a' mod m is then stored in the A' register **52**. The rightmost bit, containing w_1, of the W register **51** is then examined, as before, and the process repeats.

This process is repeated a maximum of k times or until the W register **51** contains all 0's, at which point wa' modulo m is stored in the P register **54**.

As an example of these operations, consider the problem of computing 7×7 modulo **23**. The following steps show the successive contents of the W, A' and P registers which result in the answer 7×7=3 modulo **23**.

i	W (in binary)	A'	P
0	00111	7	0
1	00011	14	0 + 7 = 7
2	00001	5	7 + 14 = 21
3	00000	10	21 + 5 = 3 mod 23

FIG. **4** depicts an implementation of an adder **42** or **55** for adding two k bit numbers p and z. The numbers are presented one bit at a time to the device, low order bit first, and the delay element is initially set to 0. (The delay represents the binary carry bit.) The AND gate **61** determines if the carry bit should be a one based on p_i and z_i both being 1 and the AND gate **62** determines if the carry should be 1 based on the previous carry being a 1 and one of p_i or z_i being 1. If either of these two conditions is met, the OR gate **63** has an output of 1 indicating a carry to the next stage. The two exclusive-or (XOR) gates **64** and **65** determine the i^{th} bit of the sum, s_i, as the modulo-2 sum of p_i, z_i and the carry bit from the previous stage. The delay **66** stores the previous carry bit. Typical parts for implementing these gates and the delay are SN7400, SN7404, and SN7474.

FIG. **5** depicts an implementation of a comparator **44** or **56** for comparing two numbers p and m. The two numbers are presented one bit at a time, high order bit first. If neither the p<m nor the p>m outputs have been triggered after the last bits p_o and m_o have been presented, then $p=m$. The first triggering of either the p<m or the p>m output causes the comparison operation to cease. The two AND gates **71** and **72** each have one input inverted (denoted by a circle at the input). An SN7400 and SN7404 provide all of the needed logic circuits.

FIG. **6** depicts an implementation of a subtractor **57** for subtracting two numbers. Because the numbers subtracted in FIG. **3** always produce a non-negative difference, there is no need to worry about negative differences. The larger number, the minuend, is labelled p and the smaller number, the subtrahend, is labelled m. Both p and m are presented serially to the subtractor **57**, low order bit first. AND gates **81** and **83**, OR gate **84** and XOR gate **82** determine if borrowing (negative carrying) is in effect. A borrow occurs if either $p_i=0$ and $m_i=1$, or $p_i=m_i$ and borrowing occurred in the previous stage. The delay **85** stores the previous borrow state. The ith bit of the difference, d_i, is computed as the XOR, or modulo-2 difference, of p_i, m_i and the borrow bit. The output of XOR gate **82** gives the modulo-2

difference between p_i and m_i, and XOR gate **86** takes the modulo-2 difference of this with the previous borrow bit. Typical parts for implementing these gates and the delay are SN7400, SN7404 and SN7474.

The deciphering device **16** is depicted in FIG. 7. It is given the ciphertext S, and the deciphering key consisting of w, m and a', and must compute x.

To compute x, first, w and m are input to a modulo m inverter **91** which computes w^{-1} mod m. It then uses the modulo m multiplier **92** to compute $S' = w^{-1} S \bmod m$. As noted in equations (7) and (8), $S' = a' * x$, which is easily solved for x. The comparator **93** then compares S' with a_n' and decides that $x_n = 1$ if $S' \geqq a_n'$ and that $x_n = 0$ if $S' < a_n'$. If $x_n = 1$, S' is replaced by $S' - a_n'$, computed by the subtractor **94**. If $x_n = 0$, S' is unchanged. The process is repeated for a_{n-1}' and x_{n-1} and continues until x is computed. The j register **95** is initially set to n and is decremented by 1 after each stage of the deciphering process until $j = 0$ results, causing a halt to the process and signifying x is computed. Either the subtractor **94** or a down counter can be used to decrement the contents of the j register **95**. The comparator **96** can be used to compare the contents of the j register **95** with zero to determine when to halt the process. The modulo m multiplier **92** is detailed in FIG. 3; the comparator **93** is detailed in FIG. 5; and, the subtractor **94** is detailed in FIG. 6. The modulo m invertor **91** can be based on a well known extended version of Euclid's algorithm. (See for instance, D. Knuth. *The Art of Computer Programming*, Vol. II, Addison-Wesley, 1969, Reading, Ma., p. 302 and p. 315 problem 15.) As described by Knuth, an implementation requires six registers, a comparator, a divider and a subtractor. All of these devices have already been detailed with the exception of the divider.

FIG. 8 details an apparatus for dividing an integer u by another integer v to compute a quotient q and a remainder r, such that $0 \leqq r \leqq v - 1$. First, u and v, represented as binary numbers, are loaded into the U and V registers **101** and **102**, respectively. Then v, the contents of the V register **102**, are shifted to the left until a 1 appears in v_{k-1}, the leftmost bit of the V register **102**. This process can be effected by using the complement of v_{k-1} to drive the shift control on a shift register, such as the Signetics 2533, which was initially set to zero. The contents of the up-down counter **103** equal the number of bits in the quotient less one.

After this initialization, v, the contents of the V register **102** are compared with the contents of the U register **101** by the comparator **104**. If $v > u$ then q_n, the most significant bit of the quotient, is 0 and u is left unchanged. If $v \leqq u$ then $q_n = 1$ and u is replaced by u-v as computed by the subtractor **105**. In either event, v is shifted to the right one bit and the $v > u$ comparison is repeated to compute q_{n-1}, the next bit in the quotient.

This process is repeated, with the up-down counter **103** being decremented by 1 after each iteration until it contains zero. At that point, the quotient is complete and the remainder r is in the U register **101**.

As an example, consider dividing 14 by 4 to produce $q = 3$ and $r = 2$ with $k = 4$ being the register size. Because $u = 14 = 1110$ and $v = 4 = 0100$ in binary form, the V register **101** is left shifted only once to produce $v = 1000$. After this initialization, it is found that $v \leqq u$ so the first quotient bit $q_1 = 1$, and u is replaced by u-v; v is replaced by v right shifted one bit and the up-down counter **103** is decremented to zero. This signals that the last quotient bit, q_0, is being computed, and that after the present iteration the remainder, r, is in the U register. The following sequence of register contents helps in following these operations.

U	V	counter	j
1110	1000	1	1
0110	0100	0	1
0010	—	halt	—

It is seen that $q = 11$ in binary form and is equivalent to $q = 3$, and that $r = 0010$ in binary form and is equivalent to $r = 2$.

Another method for generating a trap door knapsack vector a uses the fact that a multiplicative knapsack is easily solved if the vector entries are relatively prime. Given $a' = (6, 11, 35, 43, 169)$ and a partial product $P = 2838$, it is easily determined that $P = 6 * 11 * 43$ because 6, 11 and 43 evenly divide P but 35 and 169 do not. A multiplicative knapsack is transformed into an additive knapsack by taking logarithms. To make both vectors have reasonable values, the logarithms are taken over GF(m), the Galois (finite) field with m elements, where m is a prime number. It is also possible to use non-prime values of m, but the operations are somewhat more difficult.

A small example is again helpful. Taking $n = 4$, $m = 257$, $a' = (2, 3, 5, 7)$ and the base of the logarithms to be $b = 131$ results in $a = (80, 183, 81, 195)$. That is $131^{80} = 2 \bmod 257$; $131^{183} = 3 \bmod 257$; etc. Finding logarithms over GF (m) is relatively easy if $m - 1$ has only small prime factors.

Now, if the deciphering device **16** is given $S = 183 + 81 = 264$, it uses the deciphering key D consisting of m, a' and b, to compute

$$\begin{aligned} S' &= b^S \bmod m \\ &= 131^{264} \bmod 257 \\ &= 15 \\ &= 3 * 5 \\ &= a_1'^0 * a_2'^1 * a_3'^1 a_4'^0 \end{aligned} \tag{10}$$

which implies that $x = (0, 1, 1, 0)$. This is because

$$b^S = b^{(\Sigma a_i' * x_i)} \tag{11}$$

$$= \pi\, b^{(a_i' * x_i)} \tag{12}$$

$$= \pi\, a_i'^{\,x_i} \bmod m \tag{13}$$

However, it is necessary that

$$\prod_{i=1}^{n} a_i' < m \tag{14}$$

to ensure that $\pi a_i'^{xi} \bmod m$ equals $\pi a_i'^{xi}$ in arithmetic over the integers.

The eavesdropper **13** knows the enciphering key E, comprised of the vector a, but does not know the deciphering key D and faces a computationally infeasible problem.

The example given was again small and only intended to illustrate the technique. Taking $n = 100$, if each a_i' is a random, 100 bit prime number, then m would have to be approximately 10,000 bits long to ensure that (14) is met. While a 100:1 data expansion is acceptable in certain applications (e.g., secure key distribution over an insecure channel), it probably is not necessary for an opponent to be so uncertain of the a_i'. It is even possible to use the first n primes for the a_i', in which case m

could be as small as 730 bits long when $n=100$ and still meet condition (14). As a result, there is a possible tradeoff between security and data expansion.

In this embodiment, the enciphering device **15** is of the same form as detailed in FIG. **2** and described above. The deciphering device **16** of the second embodiment is detailed in FIG. **9**. The ciphertext S and part of the deciphering key D, namely b and m, are used by the exponentiator **111** to compute $P=b^S \bmod m$. As noted in equations (12) to (14) and in the example, P is a partial product of the $\{a_i\}$, also part of the deciphering key D. The divider **112** divides P by a_i' for $i=1, 2, \ldots n$ and delivers only the remainder r_i to the comparator **113**. If $r_i=0$ then a_i' evenly divides P and $x_i=1$. If $r_i \neq 0$ then $x_i=0$. The divider **112** may be implemented as detailed in FIG. **8** and described above. The comparator **113** may be implemented as detailed in FIG. **5** and described above; although, more efficient devices exist for comparing with zero.

The exponentiator **111**, for raising b to the S power modulo m, can be implemented in electronic circuitry as shown in FIG. **10**. FIG. **10** shows the initial contents of three registers **121**, **122** and **123**. The binary representation of S ($s_{k-1}s_{k-2}\ldots s_1s_0$) is loaded into the S register **121**; 1 is loaded into the R register **122**; and the binary representation of b is loaded into the B register **123**, corresponding to $i=0$. The number of bits k in each register is the least integer such that $2^k \geqq m$. If $k=200$, then all three registers can be obtained from a single 1024 bit random access memory (RAM) such as the Intel 2102. The implementation of multiplier **124**, for multiplying two numbers modulo m, has been described in detail in FIG. **3**.

Referring to FIG. **10**, if the low order bit, containing s_0, of the S register **121** equals 1 then the R register **122** and the B register **123** contents are multiplied modulo m and their product, also a k bit quantity, replaces the contents of the R register **122**. If $s_0=0$, the R register **122** contents are left unchanged. In either case, the B register **123** is then loaded twice into the multiplier **124** so that the square, modulo m, of the B register **123** contents is computed. This valve, $b^{(2i+1)}$, replaces the contents of the B register **123**. The S register **121** contents are shifted one bit to the right and a 0 is shifted in at the left so its contents are now $0s_{k-1}s_{k-2}\ldots s_2s_1$.

The low order bit, containing s_1, of the S register **121** is examined. If it equals 1 then, as before, the R register **122** and B register **123** contents are multiplied modulo m and their product replaces the contents of the R register **122**. If the low order bit equals 0 then the R register **122** contents are left unchanged. In either case, the contents of the B register **123** are replaced by the square, modulo m, of the previous contents. The S register **121** contents are shifted one bit to the right and a 0 is shifted in at the left so its contents are now $00s_{k-1}s_{k-2}\ldots s_3s_2$.

This process continues until the S register **121** contains all 0's, at which point the value of b^S modulo m is stored in the R register **122**.

An example is helpful in following this process. Taking $m=23$, we find $k=5$ from $2^k \geqq m$. If $b=7$ and $S=18$ then

$$b^S=7^{18}=1628413597910449=23(70800591213497)+18$$

so b^S modulo m equals 18. This straightforward but laborious method of computing b^S modulo m is used as a check to show that the method of FIG. **10**, shown below, yields the correct result. The R register **122** and B register **123** contents are shown in decimal form to facilitate understanding.

i	S (in binary)	R	B
0	10010	1	7
1	01001	1	3
2	00100	3	9
3	00010	3	12
4	00001	3	6
5	00000	18	13

The row marked $i=0$ corresponds to the initial contents of each register, $S=18$, $R=1$ and $B=b=7$. Then, as described above, because the right most bit of S register **121** is 0, the R register **122** contents are left unchanged, the contents of the B register **123** are replaced by the square, modulo 23, of its previous contents ($7^2=49=2\times23+3=3$ modulo 23), the contents of the S register **121** are shifted one bit to the right, and the process continues. Only when $i=1$ and 4 do the rightmost bit of the S register **121** contents equal 1, so only going from $i=1$ to 2 and from $i=4$ to 5 is the R register **122** replaced by RB modulo m. When $i=5$, $S=0$ so the process is complete and the result, **18**, is in the R register **122**.

Note that the same result, **18**, is obtained here as in the straightforward calculation of 7^{18} modulo **23**, but that here large numbers never resulted.

Another way to understand the process is to note that the B register **123** contains b, b^2, b^4, b^8 and b^{16} when $i=0$, 1, 2, 3 and 4 respectively, and that $b^{18}=b^{16}b^2$, so only these two values need to be multiplied.

The key generator **22** used in the second embodiment is detailed in FIG. **11**. A table of n small prime numbers, p_i, is created and stored in source **131**, which may be a read only memory such as the Intel **2316E**. The key source **26**, as described above, generates random numbers, e_i. The small prime numbers from the source **131** are each raised to a different power, represented by a random number e_i from key source **26**, by the exponentiator **132** to generate $p_i^{e_i}$ for $i=1$ to n. The multiplier **133** then computes the product of all the $p_i^{e_i}$ which may be represented as

$$\prod_{i=1}^{n} p_i^{e_i}.$$

The product of all the

$$p_i^{e_i} \cdot \prod_{i=1}^{n} p_i^{e_i},$$

then is incremented by one by adder **134** to generate the potential value of m. If it is desired that m be prime, the potential value of m may be tested for primeness by prime tester **135**.

Prime testers for testing a number m for primeness when the factorization of $m-1$ is known

$$\left(\text{as here, } m-1=\prod_{i=1}^{n} p_i^{e_i}\right),$$

are well documented in the literature. (See for instance, D. Knuth, *The Art of Computer Programming*, vol. II, Seminumerical Algorithms, pp. 347–48.) As described in the above reference, the prime tester **135** requires only a means for exponentiating various numbers to

various powers modulo m, as described in FIG. 10. When a potential value of m is found to be prime, it is output by the public key generator of FIG. 11 as the variable m. The a' vector's elements, a_i', can then be chosen to be the n small prime numbers, p_i, from source 131.

The base, b, of the logarithms is then selected as a random number by the key source 26.

The elements of the vector a are computed by the logarithmic convertor 136 as the logarithms, to the base b, of the elements of the a' vector over GF(m). The operation and structure of a logarithmic convertor 136 is described below.

It is well known that if p is prime then

$$z^{p-1}=1 \pmod p,\ 1\leqq z\leqq p-1 \tag{15}$$

Consequently arithmetic in the exponent is done modulo p−1, not modulo p. That is

$$z^x=z^{x(\mathrm{mod}\ p-1)} \pmod p \tag{16}$$

for all integers x.

The algorithm for computing logarithms mod p is best understood by first considering the special case $p=2^n+1$. We are given α, p and y, with α a primitive element of GF(p), and must find x such that $y=\alpha^x \pmod p$. We can assume $0\leqq x\leqq p-2$, since $x=p-1$ is indistinguishable from $x=0$.

When $p=2^n+1$, x is easily determined by finding the binary expansion $\{b_0, \ldots, b_{n-1}\}$ of x, The least significant bit, b_0, of x is determined by raising y to the $(p-1)/2=2^{n-1}$ power and applying the rule

$$y^{(p-1)/2} \pmod p = \begin{cases} +1, & b_o = 0 \\ -1, & b_o = 1. \end{cases} \tag{17}$$

This fact is established by noting that since α is primitive

$$\alpha^{(p-1)/2}=-1 \pmod p, \tag{18}$$

and therefore

$$y^{(p-1)/2}=(\alpha^x)^{(p-1)/2}=(-1)^x \pmod p. \tag{19}$$

The next bit in the expansion of x is then determined by letting

$$z=y\alpha^{-b_0}=\alpha^{x_1} \pmod p \tag{20}$$

where

$$x_1=\sum_{i=1}^{n-1} b_i 2^i. \tag{21}$$

Clearly x_1 is a multiple of 4 if and only if $b_1=0$. If $b_1=1$ then x_1 is divisible by 2, but not by 4. Reasoning as before,

$$z^{(p-1)/4} \pmod p = \begin{cases} +1, & b_1 = 0 \\ -1, & b_1 = 1. \end{cases} \tag{22}$$

The remaining bits of x are determined in a similar manner. This algorithm is summarized in the flow chart of FIG. 12.

To aid in understanding this flowchart, note that at the start of the i^{th} loop,

$$m=(p-1)/2^{i+1} \tag{23}$$

and

$$z=\alpha^{x_i} \pmod p \tag{24}$$

where

$$x_i=\sum_{j=i}^{n-1} b_j 2^j. \tag{25}$$

Thus raising z to the m^{th} power gives

$$z^m=\alpha^{(x_i m)}=\alpha^{[(p-1)/2]\cdot(x_i/2^i)} = (-1)^{x_i/2^i}=(-1)^{b_i} \pmod p, \tag{26}$$

so that $z^m=1 \pmod p$ if and only if $b_i=0$, and $z^m=-1 \pmod p$ if and only if $b_i=1$.

As an example, consider $p=17=2^4+1$. Then $\alpha=3$ is primitive ($\alpha=2$ is not primitive because $2^8=256=1 \pmod{17}$. Given $y=10$ the algorithm computes x as follows (note that $\beta=x^{-1}=6$ since $3\times6=18=1 \pmod{17}$):

i	Z	β	m	W	b_i
0	10	6	8	16	1
1	9	2	4	16	1
2	1	4	2	1	0
3	1	16	1	1	0
4		1			

It thus finds that $x=2^0+2^1=3$. This is correct because $\alpha^3=3^3=27=10 \pmod{17}$.

We now generalize this algorithm to arbitrary primes p. Let

$$p-1=p_1^{n_1}p_2^{n_2}\cdots p_k^{n_k},\ p_i<p_{i+1} \tag{27}$$

be the prime factorization of p−1, where the p_i are distinct primes, and the n_i are positive integers. The value of x (mod $p_i^{n_i}$) will be determined for $i=1, \ldots, k$ and the results combined via the Chinese remainder theorem to obtain

$$x \left(\mathrm{mod} \prod_{i=1}^{k} p_i^{n_i}\right) = x \pmod{p-1} = x \tag{28}$$

since $0\leqq x\leqq p-2$. The Chinese remainder theorem can be implemented in $0(k \log_2 p)$ operations and $0(k \log_2 p)$ bits of memory. (We count a multiplication mod p as one operation.)

Consider the following expansion of x (mod $p_i^{n_i}$).

$$x \left(\mathrm{mod}\ p_i^{n_i}\right) = \sum_{j=0}^{n_i-1} b_j p_i^j \tag{29}$$

where $0\leqq b_j\leqq p_i-1$.

The least significant coefficient, b_0, is determined by raising y to the $(p-1)/p_i$ power,

$$y^{(p-1)/p_i}=\alpha^{(p-1)x/p_i}=\gamma_i^x=(\gamma_i)^{b_0} \pmod p \tag{30}$$

where

ti $\gamma_i = \alpha^{(p-1)/p_i} \pmod p$ (31)

is a primitive p_i^{th} root of unity. There are therefore only p_i possible values for $y^{(p-1)/p_i} \pmod p$, and the resultant value uniquely determines b_0.

The next digit, b_1, in the base p_i expansion of x (mod $p_i^{n_i}$) is determined by letting

$$z = y \cdot \alpha^{-b_0} = \alpha^{x_1} \pmod p, \tag{32}$$

where

$$x_1 = \sum_{j=1}^{n_i - 1} b_j p_i^j. \tag{33}$$

Now, raising z to the $(p-1)/p_i^2$ power yields

$$z^{(p-1)/p_i^2} = \alpha^{(p-1)x_1/p_i^2} = \gamma_i^{x_1/p_i} = (\gamma_i)^{b_1} \pmod p. \tag{34}$$

Again, there are only p_i possible values of $z^{(p-1)/p_i^2}$ and this value determines b_1. This process is continued to determine all the coefficients, b_j.

The flow chart of FIG. **13** summarizes the algorithm for computing the coefficients (b_j) of the expansion (**29**). This algorithm is used k times to compute x (mod $p_i^{n_i}$) for $i = 1, 2, \ldots k$, and these results are combined by the Chinese remainder theorem to obtain x. The function $g_i(w)$ in FIG. **13** is defined by

$$\gamma_i^{g_i(w)} = w \pmod p, \; 0 \leq g_i(w) \leq p_i - 1. \tag{35}$$

where γ_i is defined in (31).

If all prime factors, $\{p_i\}_{i=1}^k$, of $p-1$ are small then the $g_i(w)$ functions are easily implemented as tables, and computing a logarithm over GF(p) requires $0(\log_2 p)^2$ operations and only minimal memory for the $g_i(w)$ tables. The dominant computational requirement is computing $w = z^n$, which requires $0(\log_2 p)$ operations. This loop is traversed

$$\sum_{i=1}^{k} n_i$$

times, and if all p_i are small,

$$\sum_{i=1}^{k} n_i$$

is approximately $\log_2 p$. Thus when $p-1$ has only small prime factors it is possible to compute logarithms over GF(p) easily.

As an example, consider $p = 19$, $\alpha = 2$, $y = 10$. Then $p - 1 = 2 \cdot 3^2$ and $p_1 = 2$, $n_1 = 1$, $p_2 = 3$ and $n_2 = 2$. The computation of $x \pmod{p_1^{n_1}} = x \pmod 2$ involves computing $y^{(p-1)/p_1} = \alpha^9 = 512 = 18 \pmod{19}$ so $b_1 = 1$ and $x \pmod 2 = 2^0 = 1$ (i.e., x is odd). Next the flow chart of FIG. **13** is re-executed for $p_2 = 3$ $n_2 = 2$ as follows ($\beta = 10$ because $2 \times 10 = 20 = 1$, mod 19; further $\gamma_2 = \alpha^6 = 7$ and $7^0 = 1$, $7^1 = 7$, and $7^2 = 11 \pmod{19}$ so $g_2(1) = 0$, $g_2(7) = 1$ and $g_2(11) = 2$):

Z	B	n	j	W	b_j
10	10	6	0	11	2
12	12	2	1	11	2
18	18	1	2		

so that $x \pmod{p_2^{n_2}} = x \pmod 9 = 2 \cdot 3^0 + 2 \cdot 3^1 = 8$.

Knowing that $x \pmod 2 = 1$ and that $x \pmod 9 = 8$ implies that $x \pmod{18} = 17$. (Either the Chinese Remainder Theorem can be used, or it can be realized that $x = 8$ or $x = 8 + 9 = 17$ and only 17 is odd.) As a check we find that $2^{17} = 131{,}072 = 10 \pmod{19}$, so that $y = \alpha^x \pmod p$.

It is seen that the logarithmic convertor requires a mod p inverter for computing $\beta = \alpha^{-1} \pmod p$. As already noted, this can be obtained using the extended form of Euclid's algorithm, which requires the use of the divider of FIG. **8**, the multiplier of FIG. **3**, and the comparator of FIG. **5**. The logarithmic convertor also requires the divider of FIG. **8** (for computing successive values of n), the adder of FIG. **4** (for incrementing j), the modulo p exponentiator of FIG. **10** (for computing W and β^{b_j} and for precomputing the $g_i(W)$ table), the modulo p multiplier of FIG. **3** (for computing successive values of Z), and the comparator of FIG. **5** (for determining when $j = N_i$). The logarithmic convertor's use of the Chinese remainder theorem requires only devices which have already been described (the multiplier of FIG. **3** and a modulo m inverter).

In the first method of generating a trap door knapsack vector, a very difficult knapsack problem involving a vector a was transformed into a very simple and easily solved knapsack problem involving a', by means of the transformation:

$$a_i' = 1/w^* a_i \bmod m \tag{36}$$

A knapsack involving a could be solved because it was transformable into another knapsack involving a' that was solvable. Notice, though, that is does not matter why knapsacks involving a' are solvable. Thus, rather than requiring that a' satisfy (1), it could be required that a' be transformable into another knapsack problem involving a'', by the transformation:

$$a_i'' = 1/w'^* a_i' \bmod m' \tag{37}$$

where a'' satisfies (1), or is otherwise easy to solve. Having done the transformation twice, there is no problem in doing this a third time; in fact, it is clear that this process may be iterated as often as desired.

With each successive transformation, the structure in the publicly known vector, a, becomes more and more obscure. In essence, we are encrypting the simple knapsack problem by the repeated application of a transformation which preserves the basic structure of the problem. The final result a appears to be a collection of random numbers. The fact that the problem can be easily solved has been totally obscured.

The original, easy to solve knapsack vector can meet any condition, such as (1) which guarantees that it is easy to solve. For example it could be a multiplicative trap door knapsack. In this way it is possible to combine both of the trap door knapsack methods into a single method, which is presumably harder to break.

It is important to consider the rate of growth of a, because this rate determines the data expansion involved in transmitting the n dimensional vector x as the larger quantity S. The rate of growth depends on the method of selecting the numbers, but in a "reasonable" implementation, with $n = 100$, each a_i will be at most 7 bits larger than the corresponding a'_i, each a'_i will be at most 7 bits larger than a''_i, etc., etc. Each successive stage of the transformation will increase the size of the

problem by only a small, fixed amount. If we repeat the transformation 20 times, this will add at most 140 bits to each a_i. If each a_i is 200 bits long to begin with, then they need only be 340 bits long after 20 stages. The knapsack vector, for $n=100$, will then be at most $100*340=34$ kilobits in size.

Usual digital authenticators protect against third party forgeries, but cannot be used to settle disputes between the transmitter 11 and receiver 12 as to what message, if any, was sent. A true digital signature is also called a receipt because it allows the receiver 12 to prove that a particular message M was sent to it by the transmitter 11. Trap door knapsacks can be used to generate such receipts in the following manner.

If every message M in some large fixed range had an inverse image x, then it could be used to provide receipts. Transmitter 11 creates knapsack vectors b' and b such that b' is a secret key, such as an easily solved knapsack vector, and that b is a public key, such as is obtained via the relation

$$b_i = w * b_i' \bmod m \tag{38}$$

Vector b is then either placed in a public file or transmitted to receiver 12. When transmitter 11 wants to provide a receipt for message M, transmitter 11 would compute and transmit x such that $b*x=M$. Transmitter 11 creates x for the desired message M by solving the easily solved knapsack problem.

$$\begin{aligned} M' &= 1/w * M \bmod m && (39)\\ &= 1/w * \Sigma x_i * b_i \bmod m && (40)\\ &= 1/w * \Sigma x_i * w * b_i' \bmod m && (41)\\ &= \Sigma x_i * b_i' \bmod m && (42) \end{aligned}$$

The receiver 12 could easily compute M from x and, by checking a date/time field (or some other redundancy in M), determine that the message M was authentic. Because the receiver 12 could not generate such an x, since it requires b' which only the transmitter 11 possesses, the receiver 12 saves x as proof that transmitter 11 sent message M.

This method of generating receipts can be modified to work when the density of solutions (the fraction of messages M between 0 and Σb_i which have solutions to $b*x=M$) is less than 1, provided it is not too small. The message M is sent in plaintext form, or encrypted as described above if eavesdropping is a concern, and a sequence of related oneway functions $y_1=F_1(M)$, $y_2=F_2(M)$, . . . are computed. The transmitter 11 then seeks to obtain an inverse image, x, for $y_1, y_2, \ldots$ until one is found and appends the corresponding x to M as a receipt. The receiver 12 computes $M'=b*x$ and checks that $M'=y_i$ where i is within some acceptable range.

The sequence of one-way functions can be as simple as:

$$F(M)=F(M)+i \tag{43}$$

or

$$F(M)=F(M+i) \tag{44}$$

where F(*) is a one-way function. It is necessary that the range of F(*) have at least 2^{100} values to foil trial and error attempts at forgery.

It is also possible to combine the message and receipt as a single message-receipt datum. If the acceptable range for i is between 0 and 2^I-1 and the message is J bits long then a single number, J+I bits long, can represent both the message and i. The transmitter 11 checks for a solution to $b*x=S$ for each of the 2^I values of S which result when, for example, the first J bits of S are set equal to the message and the last I bits of S are unconstrained. The first such solution x is conveyed to the receiver 12 as the message-receipt. Receiver 12 recovers S by computing the dot product of the public key b and the message-receipt combination x, and retaining the first J bits of S thus obtained. The authenticity of the message is validated by the presence of appropriate redundancy in the message, either natural redundancy if the message is expressed in a natural language such as English, or artificial redundancy such as the addition of a date-time field in the message.

Redundancy is used here in the sense of information theory [Claude E. Shannon, "The Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27, p. 379 and p. 623, 1948] and complexity theory [Gregory J. Chaitin "On the Length of Programs for Computing Finite Binary Sequences", Journal of the Association for Computing Machinery, Vol. 13, p. 547, 1966] to measure the structure (deviation from complete randomness and unpredictability) in a message. A source of messages possesses no redundancy only if all characters occur with equal probability. If it is possible to guess the characters of the message with a better than random success rate, the source possesses redundancy and the rate at which a hypothetical gambler can make his fortune grow is the quantitative measure of redundancy. [Thomas M. Cover and Roger C. King, "A Convergent Gambling Estimate of the Entropy of English", Technical Report #22, Statistics Department, Stanford University, Nov. 1, 1976]. Humans can easily validate the message by performing a redundancy check (e.g., determine if the message is grammatically correct English). By simulating the gambling situation, it is possible for a machine to validate whether or not a message possesses the redundancy appropriate to its claimed source.

There are many methods for implementing this form of the invention. Part of the deciphering key D could be public knowledge rather than secret, provided the part of D which is withheld prevents the eavesdropper 13 from recovering the plaintext message X.

Variations on the above described embodiment are possible. For example, in some applications, it will prove valuable to have the i^{th} receiver of the system generate a trap door knapsack vector $a^{(i)}$ as above, and place the vector or an abbreviated representation of the vector in a public file or directory. Then, a transmitter who wishes to establish a secure channel will use $a^{(i)}$ as the enciphering key for transmitting to the i^{th} receiver. The advantage is that the i^{th} receiver, once having proved his identity to the system through the use of his driver's license, fingerprint, etc., can prove his identity to the transmitter by his ability to decipher data encrypted with enciphering key $a^{(i)}$. Thus, although the best mode contemplated for carrying out the present invention has been herein shown and described, it will be apparent that modification and variation may be made without departing from what is regarded to be the subject matter of this invention.

What is claimed is:

1. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

providing random numbers at the receiver;

generating from said random numbers a public enciphering key at the receiver;

generating from said random numbers a secret deciphering key at the receiver such that the secret deciphering key is directly related to and computationally infeasible to generate from the public enciphering key;

communicating the public enciphering key from the receiver to the transmitter;

processing the message and the public enciphering key at the transmitter and generating an enciphered message by an enciphering transformation, such that the enciphering transformation is easy to effect but computationally infeasible to invert without the secret deciphering key;

transmitting the enciphered message from the transmitter to the receiver; and

processing the enciphered message and the secret deciphering key at the receiver to transform the enciphered message with the secret deciphering key to generate the message.

2. In a method of communicating securely over an insecure communication channel as in claim 1, further comprising:

authenticating the receiver's identity to the transmitter by the receiver's ability to decipher the enciphered message.

3. In a method of communicating securely over an insecure communication channel as in claim 2 wherein the step of:

authenticating the receiver's identity includes the receiver transmitting a representation of the message to the transmitter.

4. In a method of providing a digital signature for a communicated message comprising the steps of

providing random numbers at the transmitter;

generating from said random numbers a public key at the transmitter;

generating from said random numbers a secret key at the transmitter such that the secret key is directly related to and computationally infeasible to generate from the public key;

processing the message to be transmitted and the secret key at the transmitter to generate a digital signature at said transmitter by transforming a representation of the message with the secret key, such that the digital signature is computationally infeasible to generate from the public key;

communicating the public key to the receiver;

transmitting the message and the digital signature from the transmitter to the receiver;

receiving the message and the digital signature at the receiver and transforming said digital signature with the public key to generate a representation of the message; and

validating the digital signature by comparing the similarity of the message to the representation of the message generated from the digital signature.

5. A method of providing a message digital signature receipt for a communicated message comprising the steps of:

providing random numbers at the transmitter;

generating from said random numbers a public key at the transmitter:

generating from said random numbers a secret key at the transmitter such that the secret key is directly related to and computationally infeasible to generate from the public key;

processing the message and the secret key at the transmitter and generating a message-digital signature at said transmitter by transforming a representation of the message with the secret key, such that the message-digital signature is computationally infeasible to generate from the public key;

communicating the public key to the receiver;

transmitting the message-digital signature from the transmitter to the receiver;

processing the message-digital signature and the public key at the receiver and transforming the message-digital signature with the public key; and

validating the transformed message-digital signature by checking for redundancy.

6. In an apparatus for communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

means for generating random information at the receiver;

means for generating from said random information a public enciphering key at the receiver, means for generating from said random information a secret deciphering key such that the secret deciphering key is directly related to and computationally infeasible to generate from the public enciphering key;

means for communicating the public enciphering key from the receiver to the transmitter;

means for enciphering a message at the transmitter having an input connected to receive said public enciphering key, having another input connected to receive said message, and serving to transform said message with said public enciphering key, such that the enciphering transformation is computationally infeasible to invert without the secret deciphering key;

means for transmitting the enciphered message from the transmitter to the receiver; and

means for deciphering said enciphered message at the receiver having an input connected to receive said enciphered message, having another input connected to receive said secret deciphering key and serving to generate said message by inverting said transformation with said secret deciphering key.

7. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

generating a secret deciphering key at the receiver by generating an n dimensional vector a', the elements of vector a', being defined by

$$a_i' > \sum_{j=1}^{i-1} a_j \text{ for } i = 1, 2, \ldots n$$

where n is an integer;

generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = (w^* a_i' \bmod m) + km \text{ for } i = 1, 2, \ldots n$$

where m and w are large integers, w is invertible modulo m, and k is an integer;

transmitting the public enciphering key from the receiver to the transmitter;

receiving the message and the public enciphering key at the transmitter and generating an enciphered message by computing the dot product of the message, represented as a vector x with each element being 0 or 1, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x$$

transmitting the enciphered message from the transmitter to the receiver; and

receiving the enciphered message and the secret deciphering key at the receiver and transforming the enciphered message with the secret deciphering key to generate the message by computing

$$S' = 1/w^* S \bmod m$$

and letting $x_i = 1$ if and only if

$$[S' - \sum_{j=i+1}^{n} x_j * a_j] \geq a_i'$$

and letting $x_i = 0$ if

$$[S' - \sum_{j=i+1}^{n} x_j * a_j] < a_i'$$

for $i = n, n-1, \ldots 1$.

8. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

generating a secret deciphering key at the receiver by generating an n dimensional vector a', the elements of vector a' being defined by

$$a_i' > l \sum_{j=1}^{i-1} a_j$$

for i = 1, 2, . . . n

where l and n are integers;

generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = (W^* a_i' \bmod m) + km \text{ for } i = 1, 2, \ldots n$$

where m and w are large integers, w is invertible modulo m and k is an integer;

transmitting the public enciphering key from the receiver to the transmitter;

receiving the message and the public enciphering key at the transmitter and generating an enciphered message by computing the dot product of the message, represented as a vector x with each element being an integer between 0 and l, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x;$$

transmitting the enciphered message from the transmitter to the receiver; and

receiving the enciphered message and the secret deciphering key at the receiver and transforming the enciphered message with the secret deciphering key to generate the message by computing

$$S' = 1/w^* S \bmod m$$

and letting x_i be the integer part of

$$[S' - \sum_{j=i+1}^{n} x_j * a_j']/a_i'$$

for $i = n, n-1, \ldots 1$.

9. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

generating a secret deciphering key at the receiver by generating an n dimensional vector a', the elements of vector a' being relatively prime and n being an integer;

generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = \log_b a_i' \bmod m \text{ for } i = 1, 2 \ldots n$$

where b and m are large integers and m is a prime number such that

$$m > \prod_{i=1}^{n} a_i';$$

transmitting the public enciphering key from the receiver to the transmitter;

receiving the message and the public enciphering key at the transmitter and generating an enciphered message by computing the dot product of the message, represented as a vector x, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x;$$

transmitting the enciphered message from the transmitter to the receiver; and

receiving the enciphered message and the secret deciphering key at the receiver and transforming the enciphered message with the secret deciphering key to generate the message by computing

$$S' = b^S \bmod m$$

and letting $x_i = 1$ if and only if the quotient of S'/a_i is an integer and letting $x_i = 0$ if the quotient of S'/a_i is not an integer.

10. In an apparatus for communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

means for generating a secret deciphering key at the receiver by generating an n dimension vector a', the elements of vector a' being defined by

$$a_i > \sum_{j=1}^{i-1} a_j \text{ for } i = 1, 2, \ldots n$$

where n is an integer;

means for generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = (w^* a_i' \bmod m) + km \text{ for } i = 1, 2, \ldots n$$

where m and w are large integers, w is invertible modulo m, and k is an integer;

means for transmitting the public enciphering key from the receiver to the transmitter;

means, for enciphering a message at the transmitter, having an input connected to receive the public enciphering key, having another input connected to receive the message, and having an output that generates an enciphered message that is a transformation of the message with the public enciphering key by computing the dot product of the message, represented as a vector x with each element being 0 or 1, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x$$

means for transmitting the enciphered message from the transmitter to the receiver; and

means for deciphering the enciphered message at the receiver, having an input connected to receive the enciphered message, having aother input connected to receive the secret deciphering key, and having an output for generating the message by inverting the transformation with the secret deciphering key by computing

$$S' = 1/w^* S \bmod m$$

and letting $x_i = 1$ if and only if

$$[S' - \sum_{j=i+1}^{n} x_j * a_j] \geq a_i'$$

and letting $x_i = 0$ if

$$[S' - \sum_{j=i+1}^{n} x_j * a_j] < a_i'$$

for $i = n, n-1, \ldots 1$.

11. In an apparatus for communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

means for generating a secret deciphering key at the receiver by generating an n dimensional vector a', the elements of vector a' being defined by

$$a_i' > l \sum_{j=1}^{i-1} a_j \text{ for } i = 1, 2, \ldots n$$

where l and n are integers;

means for generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = (w^* a_i' \bmod m) + km \text{ for } i = 1, 2 \ldots n$$

where m and w are large integers, w is invertible modulo m, and k is an integer;

means for transmitting the public enciphering key from the receiver to the transmitter;

means, for enciphering a message at the transmitter, having an input connected to receive the public enciphering key, having another input connected to receive the message, and having an output that generates an enciphered message that is a transformation of the message with the public enciphering key by computing the dot product of the message, represented as a vector x with each element being an integer between 0 and l, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x$$

means for transmitting the enciphered message from the transmitter to the receiver; and

means for deciphering the enciphered message at the receiver, having an input connected to receive the enciphered message, having another input connected to receive the secret deciphering key, and having an output for generating the message by inverting the transformation with the secret deciphering key by computing

$$S' = 1/w^* S \bmod m$$

and letting x_i be the integer part of

$$[S' - \sum_{j=i+1}^{n} x_j * a_j]/a_i'$$

for $i = n, n-1, \ldots 1$.

12. In an apparatus for communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

means for generating a secret deciphering key at the receiver by generating an n dimensional vector a', the elements of vector a' being relatively prime and n being an integer;

means for generating a public enciphering key at the receiver by generating an n dimensional vector a, the elements of vector a being defined by

$$a_i = \log_b a_i' \bmod m \text{ for } i = 1, 2, \ldots n$$

where b and m are large integers and m is a prime number such that

$$m > \prod_{i=1}^{n} a_i';$$

means for transmitting the public enciphering key from the receiver to the transmitter;

means, for enciphering a message at the transmitter, having an input connected to receive the public enciphering key, having another input connected to receive the message, and having an output that generates an enciphered message that is a transformation of the message with the public enciphering key by computing the dot product of the message,

represented as a vector x with each element being 0 or 1, and the public enciphering key, represented as the vector a, to represent the enciphered message S being defined by

$$S = a^* x.$$

means for transmitting the enciphered message from the transmitter to the receiver; and

means for deciphering the enciphered message at the receiver, having an input connected to receive the enciphered message, having another input connected to receive the secret deciphering key, and having an output for generating the message by inverting the transformation with the secret deciphering key by computing

$$S' = b^S \bmod m$$

and letting $x_i = 1$ if and only if the quotient of S'/a_i is an integer and letting $x_i = 0$ of the quotient of S'/a_i is not an integer.

13. In an apparatus for enciphering a message that is to be transmitted over an insecure communication channel having an input connected to receive a message to be maintained secret, having another input connected to receive a public enciphering key, and having an output for generating the enciphered message, characterized by:

means for receiving the message and converting the message to a vector representation of the message;

means for receiving the public enciphering key and converting the public enciphering key to a vector representation of the public enciphering key; and

means for generating the enciphered message by computing the dot product of the vector representation of the message and the vector representation of the public enciphering key, having an input connected to receive the vector representation of the message, having another input connected to receive the vector representation of the public enciphering key, and having an output for generating the enciphered message.

14. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver the improvement characterized by:

generating a secret deciphering key at the receiver;

generating a public enciphering key at the receiver, such that the secret deciphering key is computationally infeasible to generate from the public enciphering key;

transmitting the public enciphering key from the receiver to the transmitter;

processing the message and the public enciphering key at the transmitter by computing the dot product of the message, represented as a vector, and the public enciphering key, represented as a vector, to represent the enciphered message, such that the enciphering transformation is easy to effect but computationally infeasible to invert without the secret deciphering key;

transmitting the enciphered message from the transmitter to the receiver;

and processing the enciphered message and the secret deciphering key at the receiver and inverting said transformation by transforming the enciphered message with the secret deciphering key to generate the message.

15. In an apparatus for communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

means for generating a secret deciphering key at the receiver;

means for generating a public enciphering key at the receiver, such that the secret deciphering key is computationally infeasible to generate from the public enciphering key;

means for transmitting the public enciphering key from the receiver to the transmitter;

means for enciphering a message at the transmitter having an input connected to receive said public enciphering key and having another input connected to receive said message and serving to transform said message by computing the dot product of said message, represented as a vector, and said public enciphering key, represented as a vector, to represent said enciphered message, such that the enciphering transformation is computationally infeasible to invert without the secret deciphering key;

means for transmitting the enciphered message from the transmitter to the receiver;

and means for deciphering said enciphered message at the receiver, said means having an input connected to receive said enciphered message and having another input connected to receive said secret deciphering key and serving to generate said message by inverting the transformation with said secret deciphering key.

16. An apparatus for deciphering an enciphered message that is received over an insecure communication channel including

means for receiving the enciphered message that is enciphered by an enciphering transformation in which a message to be maintained secret is transformed with a public enciphering key, and means for receiving a secret deciphering key to generate the message by inverting the enciphering transformation;

means for generating the message having an input connected to receive the inverse of the enciphered message and an output for generating the message;

said secret deciphering key being computationally infeasible to generate from the public enciphering key, and said enciphering transformation being computationally infeasible to invert without the secret deciphering key in which

said means for inverting the enciphering transformation includes means for computing

$$S' = 1/w^* S \bmod m; \text{ and}$$

said means for generating the message includes means for setting x_i equal to the integer part of

$$\left[S' - \sum_{j=i+1}^{n} x_j * a_j \right] / a_i \text{ for } i = n, n-1, \ldots 1$$

where m and w are large integers and w is invertible modulo m, where S' is the inverse of the enciphered message S being defined by the enciphering transformation

$S=a^*x$

where the message is represented as an n dimensional vector x with each element x_i being an integer between 0 and l, where l is an integer, and where the public enciphering key is represented as an n dimensional vector a, the elements of a being defined by

$a_i=(w^*a_i' \bmod m)+km$ for $i=1, 2, \ldots n$

where k and n are integers and the secret deciphering key is m, w and a', where a' is an n dimensional vector, the elements of a' being defined by

$a_i' > l\sum_{j=1}^{i-1} a_j$ for $i = 1, 2, \ldots n$

17. An apparatus for deciphering an enciphered message that is received over an insecure communication channel including

means for receiving the enciphered message that is enciphered by an enciphering transformation in which a message to be maintained secret is transformed with a public enciphering key, and means for receiving a secret deciphering key to generate the messge by inverting the enciphering transformation;

means for generating the message having an input connected to receive the inverse of the enciphered message and an output for generating the message;

said secret deciphering key being computationally infeasible to generate from the public enciphering key, and said enciphering transformation being computationally infeasible to invert without the secret deciphering key in which

said means for inverting the enciphering transformation includes means for computing

$S'=b^S \bmod m$; and

said means for generating the message includes means for setting $x_i=1$ if and only if the quotient of S'/a_i is an integer and setting $x_i=0$ of the quotient of S'/a_i is not an integer, where b and m are large integers and m is a prime number such that

$m > \prod_{i=1}^{n} a_i'$

where n is an integer and the secret deciphering key is b,m, and a', where a' is an n dimensional vector with each element a" being relatively prime, and where S' is the inverse of the enciphered message S being defined by the enciphering transformation

$S=e^*x$

where the message is represented as an n dimensional vector x with each element x_i being 0 or 1, and the public enciphering key is represented as the n dimensional vector a, the elements of a being defined by

$a_i=\log_b a_i' \bmod m$ for $i=1, 2, \ldots n$

* * * * *

D

KAHN ON CODES

Secrets of the New Cryptology

by DAVID KAHN

MACMILLAN PUBLISHING COMPANY
New York

To Oliver and to Michael

Macmillan Publishing Company
866 Third Avenue, New York, N.Y. 10022
Collier Macmillan Canada, Inc.

Library of Congress Cataloging in Publication Data

Kahn, David, 1930–
Kahn on codes.

Includes bibliographical references and index.
1. Cryptography—Addresses, essays, lectures.
I. Title.
Z103.K29 1983 001.54'36 83–16213
ISBN 0–02–560640–9

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

computer scientists in many firms and universities began studying it; the D.E.S. is a product of this interest. Very rapidly the quantity and quality of information on cryptology being circulated outside of government channels exceeded by far what it had ever been before.

The expansion was accelerated by two Stanford University scientists' development of public-key cryptography, the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance.[15] Unlike standard cryptosystems, such as the D.E.S., in which the same key serves both to encrypt a message and to decrypt it, public-key cryptography employs one key to encrypt and another to decrypt. The two keys are mathematically related to one another, and each user possesses a pair. Each makes one key public. The other he keeps secret. Suppose user A wants to communicate secretly with user B. He looks up B's public key and encrypts his message to B in it. B applies his private key to decrypt the message. Thus anyone can send B a secret message, but only he can read it. This asymmetry can eliminate one of the most vexatious problems in practical cryptography: distributing keys to a correspondent before secret communication can be started with him. And a twist makes possible what has never been possible before with electronic messages: unforgeable signatures.

The seeming impossibility of these schemes, their boldness, and their elegance have attracted numbers of first-rate mathematicians to cryptology. There is now, for the first time, an informal network of scientists who can do sophisticated mathematical cryptology and who bounce ideas off one another in the way that advances a study rapidly and rationally.

Suddenly the nation is faced with a problem it has never had before—an information explosion in cryptology. N.S.A. worries that any mention of codebreaking might make other nations change their codes, losing intelligence and forcing the agency to redo much of its work. This happens far less often than the agency likes to think. In 1941, for example, Japan did not change its principal diplomatic cipher despite an unequivocal report that the United States had broken it. Nor did the German Navy alter its systems in World War II, despite much suspicion. Several of the countries named by the defectors Martin and Mitchell in 1960 as having had their codes broken by N.S.A. did not change them thereafter. But more cautious nations do replace their cryptosystems upon suspicion of solution, and N.S.A. fears that all the new activity in cryptology may not only dry up the flow of foreign intelligence but also inadvertently expose principles used in American ciphers. All of this has caused it to ask whether the right of unrestricted inquiry is worth the national security losses. The issue has surfaced in three recent episodes.

E

RSA Data Security, Inc.

BSAFE

A Cryptographic Toolkit

User's Manual

RSA
DATA SECURITY, INC.
THE KEYS TO PRIVACY AND AUTHENTICATION

VERSION 2.1

RSA-SW-002432

Public-Key Cryptography

In 1976, Stanford graduate student Whitfield Diffie and Stanford professor Martin Hellman invented public-key cryptography. In this system, each person owns a pair of keys, called the public key and the private key. The key pair's owner publishes the public key and keeps the private key secret.

Suppose Alice wants to send a message to Bob. She finds his public key and encrypts her message using that public key. Unlike symmetric-key cryptography, though, that key, the key used to encrypt, will not decrypt the message. Knowledge of Bob's public key will not help an eavesdropper. To decrypt, Bob uses his private key. If Bob wants to respond to Alice, he will encrypt his message using her public key.

To get a flavor of this idea, think of taking a number to a power. For instance, given values x and y, compute $z = x^y$. To "recover" x, you would not compute z^y, but rather $z^{1/y}$. You end up with the original x, since $z^{1/y} = (x^y)^{1/y} = x^{y \cdot 1/y} = x^1 = x$. You need two values to perform this exercise, a "public key," y, to compute the "encrypted value," and the inverse of the public key, or a "private key," $1/y$, to recover the original value.

This example, of course, is not practical since if you made y public, anyone could easily compute $1/y$ and know your private key. Therefore, a good public-key cryptosystem would rely on a key pair for which it is impossible (or at least intractable) to derive the private key from the public key.

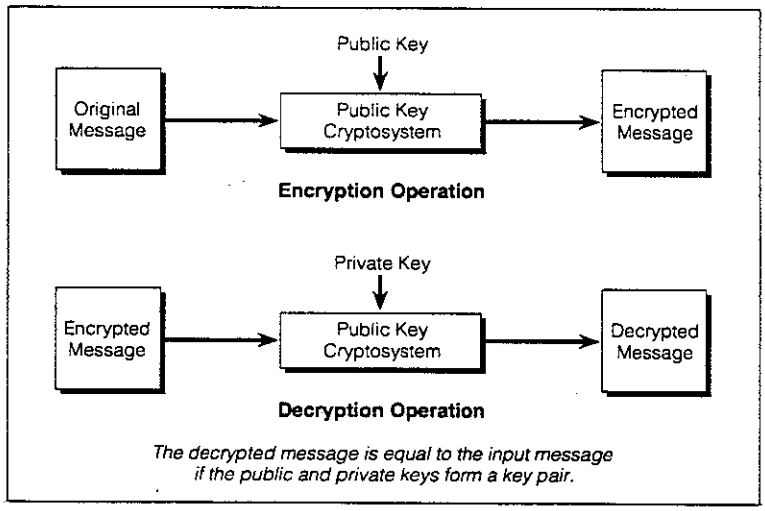


Figure 2.7 Public-Key Cryptography

When you ask for a particular individual's public key, the CA will send the certificate and, as a digital signature, the digest of the certificate encrypted with the CA's private key. To verify that the certificate is genuine, digest the certificate and decrypt the signature using the CA's public key. Compare the two results, if they are the same, you have a proper certificate.

If the CA you deal with does not have a certificate for the individual in question, that CA can talk to another CA which may have the right certificate. In fact, to find a particular certificate, a CA may have to go through a chain of CA's until it finds one that possesses the desired certificate.

Names that uniquely distinguish users are necessary for digital certificates to be of real use. The CCITT X.500 series of documents offer more discussion regarding naming conventions and related topics.

Diffie-Hellman Public Key Agreement

Whitfield Diffie and Martin Hellman invented this, the first true public-key algorithm, in 1976. It provides for key agreement, but not encryption or authentication.

The Diffie-Hellman key agreement algorithm provides a method for two parties to each compute the same secret key without exchanging secret information. Its security relies on the difficulty of computing discrete logarithms modulo a prime number. It takes very little time to exponentiate modulo a prime number, but much more to compute the inverse, the discrete logarithm. The RSA Laboratories publication, "Frequently Asked Questions About Today's Cryptography," declares, "The best discrete log problems have expected running times similar to that of the best factoring algorithms." That is, the time it takes to compute discrete logs modulo a prime of a certain length is about equivalent to the time it takes to factor a number of that same length. See the section titled "The RSA Algorithm" for a discussion on factoring.

The Diffie-Hellman algorithm is made up of three parts, Parameter Generation, Phase 1 and Phase 2.

Parameter Generation

A central authority selects a prime number p of length k bytes ($k * 8$ bits), and an integer $g \in (0, p)$, called the base. The central authority may optionally select an integer l, the private value length in bits, that satisfies $2^{l-1} \leq p$.

F

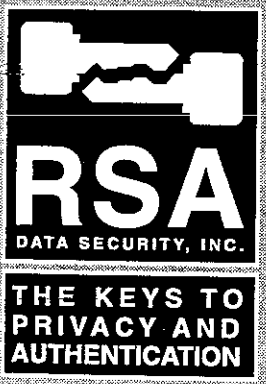
RSA Laboratories

Answers to

Frequently Asked Questions About Today's Cryptography

An introduction to modern cryptography, including answers to commonly asked questions about public key algorithms such as RSA, ElGamal and Diffie-Hellman; secret key techniques such as DES, RC2 and RC4; and hash functions such as MD, MD2, MD5 and SHA. Certificates, key management, patents, Kerberos, discrete log, factoring, domestic and international standards are also among the topics discussed.

New in this edition is expanded treatment of recent government involvement in encryption policy and standards, including discussions on the controversial Capstone, Clipper and DSS proposals, export controls, NIST, NSA, privacy and intellectual property concerns.



REVISION 2.0

DFD

method. But public-key cryptography can share the burden with secret-key cryptography to get the best of both worlds.

For encryption, the best solution is to combine public- and secret-key systems in order to get both the security advantages of public-key systems and the speed advantages of secret-key systems. The public-key system can be used to encrypt a secret key which is then used to encrypt the bulk of a file or message. This is explained in more detail in Question 2.12 in the case of RSA. *Public-key cryptography is not meant to replace secret-key cryptography, but rather to supplement it, to make it more secure.* The first use of public-key techniques was for secure key exchange in an otherwise secret-key system [29]; this is still one of its primary functions.

Secret-key cryptography remains extremely important and is the subject of much ongoing study and research. Some secret-key encryption systems are discussed in Questions 5.1 and 5.5.

1.5 Is cryptography patentable in the U.S.?

Cryptographic systems are patentable. Many secret-key cryptosystems have been patented, including DES (see Question 5.1). The basic ideas of public-key cryptography are contained in U.S. Patent 4,200,770, by M. Hellman, W. Diffie, and R. Merkle, issued 4/29/80 and in U.S. Patent 4,218,582, by M. Hellman and R. Merkle, issued 8/19/80; similar patents have been issued throughout the world. The exclusive licensing rights to both patents are held by Public Key Partners (PKP), of Sunnyvale, California, which also holds the rights to the RSA patent (see Question 2.19). Usually all of these public-key patents are licensed together.

All legal challenges to public-key patents have been settled before judgment. In a recent case, for example, PKP brought suit against the TRW Corporation which was using public-key cryptography (the ElGamal system) without a license; TRW claimed it did not need to license. In June 1992 a settlement was reached in which TRW agreed to license to the patents.

Some patent applications for cryptosystems have been blocked by intervention by the NSA (see Question 7.3) or other intelligence or defense agencies, under the authority of the Invention Secrecy Act of 1940 and the National Security Act of 1947; see Landau [46] for some recent cases related to cryptography.

1.6 Is cryptography exportable from the U.S.?

All cryptographic products need export licenses from the State Department, acting under authority of the International Traffic in Arms Regulation (ITAR), which defines cryptographic devices, including software, as munitions. The U.S. government has historically been reluctant to grant export licenses for encryption products stronger than some basic level (not publicly stated).

Under current regulations, a vendor seeking to export a product using cryptography first submits an request to the State Department's Defense Trade Control office. Export jurisdiction may then be passed to the Department of Commerce, whose export procedures are generally simple and efficient. If jurisdiction remains with the State Department, further review, perhaps lengthy, is required before export is either approved or denied; the National Security Agency (NSA, see Question 7.3) may become directly involved at this point. The details of the export approval process change frequently.

The NSA has *de facto* control over export of cryptographic products. The State Department will not grant a license without NSA approval and routinely grants licenses whenever NSA does approve. Therefore, the policy decisions over exporting cryptography ultimately rest with the NSA.